

Optimal Deterministic Polynomial-Time Data Exchange for Omniscience

Nebojsa Milosavljevic, Sameer Pawar, Salim El Rouayheb, Michael Gastpar,[†]
and Kannan Ramchandran

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

Email: {nebojsa, spawar, salim, gastpar, kannanr}@eecs.berkeley.edu

Abstract

We study the problem of constructing a deterministic polynomial time algorithm that achieves omniscience, in a rate-optimal manner, among a set of users that are interested in a common file but each has only partial knowledge about it as side-information. Assuming that the collective information among all the users is sufficient to allow the reconstruction of the entire file, the goal is to minimize the (possibly weighted) amount of bits that these users need to exchange over a noiseless public channel in order for all of them to learn the entire file. Using established connections to the multi-terminal secrecy problem, our algorithm also implies a polynomial-time method for constructing a maximum size secret shared key in the presence of an eavesdropper.

We consider the following types of side-information settings: (i) side information in the form of uncoded fragments/packets of the file, where the users' side-information consists of subsets of the file; (ii) side information in the form of linearly correlated packets, where the users have access to linear combinations of the file packets; and (iii) the general setting where the the users' side-information has an arbitrary (i.i.d.) correlation structure. Building on results from combinatorial optimization, we provide a polynomial-time algorithm (in the number of users) that, first finds the optimal rate allocations among these users, then determines an explicit transmission scheme (*i.e.*, a description of which user should transmit what information) for cases (i) and (ii).

[†]Also with the School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland.

This research was funded by the NSF grants (CCF-0964018, CCF-0830788), a DTRA grant (HDTRA1-09-1-0032), and in part by an AFOSR grant (FA9550-09-1-0120).

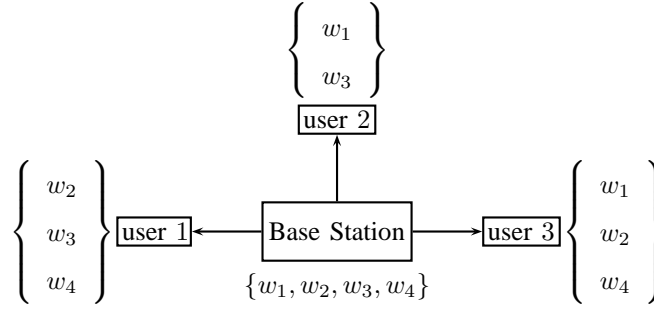


Fig. 1. An example of the data exchange problem. A base station has a file formed of four packets $w_1, \dots, w_4 \in \mathbb{F}_q$ and wants to deliver it to three users over an unreliable wireless channel. The base station stops transmitting once the users collectively have all the packets, but may individually have only subsets of the packets. For instance, here the base station stops after user 1, user 2 and user 3 have respectively packets $\{w_2, w_3, w_4\}$, $\{w_1, w_3\}$, and $\{w_1, w_2, w_4\}$, which can now be regarded as side information. The users can then cooperate among themselves to recover their missing packets. Here, the 3 users can reconcile their file with the following optimal scheme that minimizes the total amount of communicated bits: user 1 transmits packet w_4 , user 2 transmits $w_1 + w_3$, and user 3 transmits w_2 , where the addition is in the field \mathbb{F}_q .

I. INTRODUCTION

In the recent years cellular systems have witnessed significant improvements in terms of data rates and are nearly approaching theoretical limits in terms of the physical layer spectral efficiency. At the same time the rapid growth in the popularity of data-enabled mobile devices, such as smart phones and tablets, far beyond the early adoption stage, and correspondingly the increasing demand for more throughput are challenging our ability to meet this demand even with the current highly efficient cellular systems. One of the major bottlenecks in scaling the throughput with the increasing number of mobile devices is the last mile wireless link between the base station and the mobile devices – a resource that is shared among many users served within the cell. This motivates investigating new ways where cell phone devices can possibly cooperate among themselves to get the desired data in a peer-to-peer fashion without solely relying on the base station.

An example of such a setting is shown in Figure 1, where a base station wants to deliver the same file to multiple geographically-close users over an unreliable wireless downlink. Such scenario may occur for instance when co-workers are using their tablets to share and update files stored in the cloud (e.g., Dropbox), or when users, in the subway or a mall, are interested in watching the same popular video. For our example, let us suppose that the file consists of four equally sized packets w_1, w_2, w_3 and w_4 belonging to some finite field \mathbb{F}_q . Also, suppose that after few initial transmission attempts by the base station, the three users individually receive only parts of the file (see Figure 1), but collectively have the

entire file. Now, if the mobile users are in close vicinity and can communicate with each other, then, it is much more desirable and efficient, in terms of resource usage, to reconcile the file among users by letting them “talk” to each other without involving the base station. This cooperation has the following advantages:

- The connection to the base station is either unavailable after the initial phase of transmission, or it is too weak to meet the delay requirement.
- Transmissions within the close group of users is much more reliable than from any user to the base station due to geographical proximity.
- Local communication among users has a smaller footprint in terms of interference, thus allowing one to use the shared resources (code, time or frequency) freely without penalizing the base station’s resources, *i.e.*, higher resource reuse factor.

The problem of reconciling a file among multiple wireless users having parts of it while minimizing the cost in terms of the total number of bits exchanged is known in the literature as the *data exchange problem* and was introduced by El Rouayheb *et al.* in [1]. In terms of the example considered here, if the 3 users transmit R_1, R_2 and R_3 bits to reconcile the entire file, the data exchange problem would correspond to minimizing the sum-rate $R_1 + R_2 + R_3$ such that, when the communication is over, all the users can recover the entire file. It can be shown here that the minimum sum-rate required to reconcile the file is equal to 3 and can be achieved by the following coding scheme: user 1 transmits packet w_1 , user 2 transmits $w_1 + w_3$, and user 3 transmits w_2 , where the addition is over the underlying field \mathbb{F}_q . This corresponds to the optimal rate allocation $R_1 = R_2 = R_3 = 1$ symbol in \mathbb{F}_q .

In a subsequent work, Sprinston *et al.* [2] proposed a randomized algorithm that with *high probability* achieves the minimum number of transmissions, given that the field size \mathbb{F}_q is large enough. Courtade *et al.* [3] and Tajbakhsh *et al.* [4] formulated this problem as a linear program (LP) and showed that the proposed LP under some additional assumption¹, can be solved in polynomial time. In a more general setting, one can consider minimizing a different cost function, a “weighted sum rate”, *i.e.*, minimizing $\alpha_1 R_1 + \alpha_2 R_2 + \alpha_3 R_3$, for some non-negative weights $0 \leq \alpha_i < \infty$, $i = 1, 2, 3$, to accommodate the scenario when transmissions from different users have different costs. This problem was studied by Ozgul *et al.* [5], where the authors proposed a randomized algorithm that achieves this goal with *high probability* provided that the underlying field size is large enough.

The results above consider only the simple form of the side-information where different users observe

¹If users are allowed to split the packets into arbitrary number of smaller chunks.

partial uncoded “raw” packets/fragments of the original file. Typically, content distribution networks use coding, such as Fountain codes or linear network codes, to improve the system efficiency. In such scenarios, the side-information representing the partial knowledge gained by the users would be coded and in the form of linear combinations of the original file packets, rather than the raw packets themselves. The previous two cases of side information (“raw” and coded) can be regarded as special cases of the more general problem where the side-information has arbitrary correlation among the observed data of different users and where the goal is to minimize the weighted total communication (or exchange) cost to achieve omniscience. In [6] Csiszár and Narayan pose a related security problem referred to as the “multi-terminal key agreement” problem. They show that achieving omniscience in minimum number of bits exchanged over the public channel is sufficient to maximize the size of the shared secret key. This result establishes the connection between the Multi-party key agreement and the Data exchange problems. The authors in [6] solve the key agreement problem by formulating it as a linear program (LP) with an exponential number of rate-constraints, corresponding to all possible cut-sets that need to be satisfied, which has exponential complexity.

In this paper, we make the following contributions. First, we provide a *deterministic polynomial time* algorithm² for finding an optimal rate allocation, w.r.t. a linear weighted sum-rate cost, that achieves omniscience among users with arbitrarily correlated side information. For the data exchange problem, this algorithm computes the optimal rate allocation in polynomial time for the case of linearly coded side information (including the “raw” packets case) and for the general linear cost functions (including the sum-rate case). Moreover, for the “multi-terminal key agreement” security problem of [6], this algorithm computes the secret key capacity (maximum key length) in polynomial time. Second, for the data exchange problem, with raw or linearly coded side-information, we provide efficient methods for constructing linear network codes that can achieve omniscience among the users at the optimal rates with finite block lengths and zero-error.

The rest of the paper is organized as follows. In Section II, we describe the model and formulate the communication problem. Section III provides the necessary mathematical background in combinatorial optimization that will be needed for constructing our algorithm. In Section IV, we describe the polynomial time algorithm which finds an optimal rate allocation that minimizes the sum-rate (non-weighted case).

²The complexity of our proposed algorithm is $\mathcal{O}(m^2 \cdot SFM(m))$, where m is the number of users and $SFM(m)$ is the complexity of submodular function minimization. To the best of our knowledge, the fastest algorithm for SFM is given by Orlin in [7], and has complexity $\mathcal{O}(m^5 \cdot \gamma + m^6)$, where γ is complexity of computing the submodular function.

In Section V, we use the results of Section IV as a key building block to construct an efficient algorithm for an arbitrary linear communication cost function. In Section VI, we propose a polynomial time code construction for the data exchange problem using results in network coding. We conclude our work in Section VII.

II. SYSTEM MODEL AND PRELIMINARIES

In this paper, we consider a set up with m user terminals that are interested in achieving omniscience of a particular file or a random process. Let $X_1, X_2, \dots, X_m, m \geq 2$, denote the components of a discrete memoryless multiple source (DMMS) with a given joint probability mass function. Each user terminal $i \in \mathcal{M} \triangleq \{1, 2, \dots, m\}$ observes n i.i.d. realizations of the corresponding random variable X_i . The final goal is for each terminal in the system to gain access to all other terminals' observations, *i.e.*, to become omniscient about the file or DMMS. In order to achieve this goal the terminals are allowed to communicate over a noiseless public broadcast channel in multiple rounds and thus, may use interactive communication, meaning that transmission by a user terminal at any particular time can be a function of its initial observations as well as the past communication so far over the public broadcast channel. In [6], Csiszár and Narayan showed that to achieve the omniscience in a multi-terminal setup with general DMMS *interactive communication is not needed*. As a result, in the sequel WLOG we can assume that the transmission of each terminal is only a function of its own initial observations. Let $F_i := f_i(X_i^n)$ represent the transmission of the terminal $i \in \mathcal{M}$, where $f_i(\cdot)$ is any desired mapping of the observations X_i^n . For each terminal to achieve omniscience, transmissions $F_i, i \in \mathcal{M}$, should satisfy,

$$\lim_{n \rightarrow \infty} \frac{1}{n} H(X_{\mathcal{M}}^n | \mathbf{F}, X_i^n) = 0, \quad \forall i \in \mathcal{M}, \quad (1)$$

where $X_{\mathcal{M}} = (X_1, X_2, \dots, X_m)$.

Definition 1. A rate tuple $\mathbf{R} = (R_1, R_2, \dots, R_m)$ is an *achievable communication for omniscience (CO) rate tuple* if there exists a communication scheme with transmitted messages $\mathbf{F} = (F_1, F_2, \dots, F_m)$ that satisfies (1), *i.e.*, achieves omniscience, and is such that

$$R_i = \lim_{n \rightarrow \infty} \frac{1}{n} H(F_i), \quad \forall i \in \mathcal{M}. \quad (2)$$

In the omniscience problem every terminal is a potential transmitter as well as a receiver. As a result, any set $\mathcal{S} \subset \mathcal{M}, \mathcal{S} \neq \mathcal{M}$, defines a cut corresponding to the partition between two sets \mathcal{S} and $\mathcal{S}^c = \mathcal{M} \setminus \mathcal{S}$. It is easy to show using cut-set bounds that all the achievable CO rate tuple's necessarily belong to the

following region

$$\mathcal{R} \triangleq \{\mathbf{R} : R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}), \mathcal{S} \subset \mathcal{M}\}, \quad (3)$$

where $R(\mathcal{S}) = \sum_{i \in \mathcal{S}} R_i$. Also, using a random coding argument, it can be shown that the rate region \mathcal{R} is an achievable rate region [6]. In [8] and [9] the authors provide explicit structured codes based on syndrome decoding that achieve the rate region for a Slepian-Wolf distributed source coding problem. This approach was further extended in [10] to a multiterminal setting.

In this work, we aim to design a polynomial complexity algorithm that achieves omniscience among all the users while simultaneously minimizing an appropriately defined cost function over the rates. In the sequel we focus on the linear cost functions of the rates as an objective of the optimization problem. To that end, let $\underline{\alpha} \triangleq (\alpha_1, \dots, \alpha_m)$, $0 \leq \alpha < \infty$, be an m -dimensional vector of non-negative finite weights. We allow α_i 's to be arbitrary non-negative constants, to account for the case when communication of some group of terminals is more expensive compared to the others, *e.g.*, setting α_1 to be a large value compared to the other weights minimizes the rate allocated to the terminal 1. This goal can be formulated as the following linear program which hereafter we denote by $\text{LP}_1(\underline{\alpha})$:

$$\min \sum_{i=1}^m \alpha_i R_i, \quad \text{s.t.} \quad \mathbf{R} \in \mathcal{R}, \quad (4)$$

We use $\mathcal{R}(\underline{\alpha})$ to denote the rate region of all minimizers of the above LP, and $R_{CO}(\underline{\alpha})$ to denote the minimal cost.

Data Exchange Problem with linear correlation among users observations

As mentioned in Section I efficient content distribution networks use coding such as fountain codes or linear network codes. This results in users' observations to be in the form of linear combinations of the original packets forming the file, rather than the raw packets themselves as is the case in conventional 'Data Exchange problem'. This linear correlation source model is known in literature as *Finite linear source* [11].

Next, we briefly describe the finite linear source model. Let q be some power of a prime. Consider the N -dimensional random vector $\mathbf{W} \in \mathbb{F}_{q^n}^N$ whose components are independent and uniformly distributed over the elements of \mathbb{F}_{q^n} . Then, in the linear source model, the observation of i^{th} user is simply given by

$$\mathbf{X}_i = \mathbf{A}_i \mathbf{W}, \quad i \in \mathcal{M}, \quad (5)$$

where $\mathbf{A}_i \in \mathbb{F}_q^{\ell_i \times N}$ is an observation matrix³ for the user i .

It is easy to verify that for the finite linear source model,

$$\frac{H(X_i)}{\log q^n} = \text{rank}(\mathbf{A}_i). \quad (6)$$

Henceforth for the finite linear source model we will use the entropy of the observations and the rank of the observation matrix interchangeably.

For the sake of brevity we use the following notation

$$\text{rank} \left\{ \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \right\} \triangleq \text{rank}(\mathbf{A}, \mathbf{B}), \quad (7)$$

$$\text{rank}(\mathbf{A}|\mathbf{B}) \triangleq \text{rank}(\mathbf{A}, \mathbf{B}) - \text{rank}(\mathbf{B}). \quad (8)$$

Similar to the general DMMS model, for the finite linear source model an omniscience achievable rate tuple necessarily belongs to

$$\mathcal{R}_{de} \triangleq \{\mathbf{R} : R(\mathcal{S}) \geq \text{rank}(\mathbf{A}_{\mathcal{S}}|\mathbf{A}_{\mathcal{S}^c}), \mathcal{S} \subset \mathcal{M}\}, \quad (9)$$

where $R(\mathcal{S}) = \sum_{i \in \mathcal{S}} R_i$, and $\mathbf{A}_{\mathcal{S}}$ is a matrix obtained by stacking $\mathbf{A}_i, \forall i \in \mathcal{S}$. The rate $R_i, i \in \mathcal{M}$ is the number of symbols in \mathbb{F}_{q^n} user i transmits over the noiseless broadcast channel.

III. OPTIMIZATION OVER POLYHEDRONS AND EDMOND'S ALGORITHM

In this section we review results and techniques from the theory of combinatorial optimization. These results will form a key ingredient in finding a polynomial time algorithm for solving the rate minimization problem $\text{LP}_1(\underline{\alpha})$ which will be described in Sections IV and V. The idea is to recast the underlying rate region \mathcal{R} , defined by the cut-set constraints in (3), as a polyhedron of some set function whose dual is *intersecting submodular* which can be optimized in polynomial time. Then, we identify conditions under which the optimization problem over the dual polyhedron and the original problem have the same optimal solution.

Here, we state the definitions, theorems and algorithms that will be needed in the next sections. For a comprehensive exposition of combinatorial optimization, we refer the interested reader to references [12], [13].

³The entries in the observation matrix $A_i, \forall i \in \mathcal{M}$ denote the coefficients of the code, e.g., Fountain code or linear network code, used by the base station and hence belong to the smaller field \mathbb{F}_q rather than the field \mathbb{F}_{q^n} to which the data packets belong. This assumption is justified since the coding coefficients are typically stored in the packet in an overhead of size negligible compared to the packet length.

Definition 2 (Polyhedron). Let f be a real function defined over the set $\mathcal{M} = \{1, 2, \dots, m\}$, i.e., $f : 2^{\mathcal{M}} \rightarrow \mathbb{R}$ such that $f(\emptyset) = 0$, where $2^{\mathcal{M}}$ is the power set of \mathcal{M} . Let us define the *polyhedron* $P(f, \leq)$ and the *base polyhedron* $B(f, \leq)$ of f as follows.

$$P(f, \leq) \triangleq \{\mathbf{Z} \mid \mathbf{Z} \in \mathbb{R}^m, \quad \forall S \subseteq \mathcal{M} : Z(S) \leq f(S)\}, \quad (10)$$

$$B(f, \leq) \triangleq \{\mathbf{Z} \mid \mathbf{Z} \in P(f, \leq), \quad Z(\mathcal{M}) = f(\mathcal{M})\}, \quad (11)$$

where $Z(\mathcal{S}) = \sum_{i \in \mathcal{S}} Z_i$.

Example 1. Consider the function f defined over set $\mathcal{M} = \{1, 2\}$ such that $f(\emptyset) = 0$, $f(\{1\}) = 4$, $f(\{2\}) = 3$, and $f(\{1, 2\}) = 6$. The polyhedron $P(f)$ is defined by the region $Z_1 \leq 4$, $Z_2 \leq 3$, and $Z_1 + Z_2 \leq 6$ (see Figure 2). For the base polyhedron there is the additional constraint $Z_1 + Z_2 = 6$.

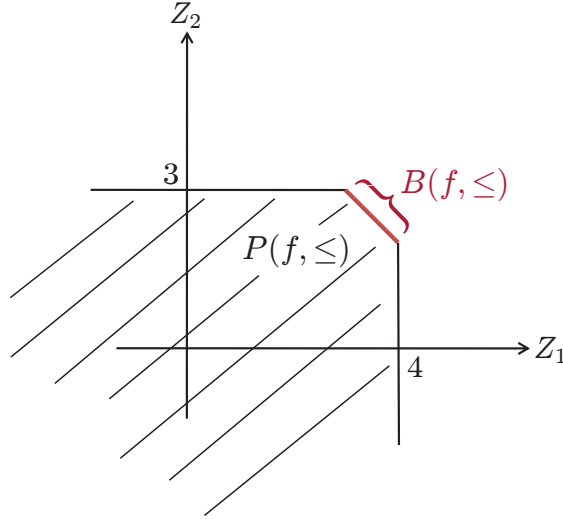


Fig. 2. Polyhedron $P(f, \leq)$ and the base polyhedron $B(f, \leq)$ for the function f specified in Example 1.

Notice that the base polyhedron $B(f, \leq)$ can be an empty set of vectors in general. For instance, if function f in Example 1 is such that $f(\{1, 2\}) = 8$ instead of 6.

Definition 3 (Dual function). For a set function f let us define its *dual function* $f^* : 2^{\mathcal{M}} \rightarrow \mathbb{R}$ as follows

$$f^*(\mathcal{S}^c) = f(\mathcal{M}) - f(\mathcal{S}), \quad \forall \mathcal{S} \subseteq \mathcal{M}, \quad (12)$$

where $\mathcal{S}^c = \mathcal{M} \setminus \mathcal{S}$.

With the dual function f^* , we associate its polyhedron and base polyhedron as follows

$$P(f^*, \geq) \triangleq \{\mathbf{R} \mid \mathbf{R} \in \mathbb{R}^m, \forall S \subseteq \mathcal{M} : R(S) \geq f^*(S)\}, \quad (13)$$

$$B(f^*, \geq) \triangleq \{\mathbf{R} \mid \mathbf{R} \in P(f^*, \geq), R(\mathcal{M}) = f^*(\mathcal{M})\}, \quad (14)$$

Lemma 1. *If $B(f, \leq) \neq \emptyset$ then, $B(f, \leq) = B(f^*, \geq)$ and $(f^*)^* = f$.*

Proof of Lemma 1 is provided in Appendix A. For the set function f from Example 1, the polyhedron $P(f^*, \geq)$ and the base polyhedron $B(f^*, \geq)$ are presented in Figure 3. We say that two optimization

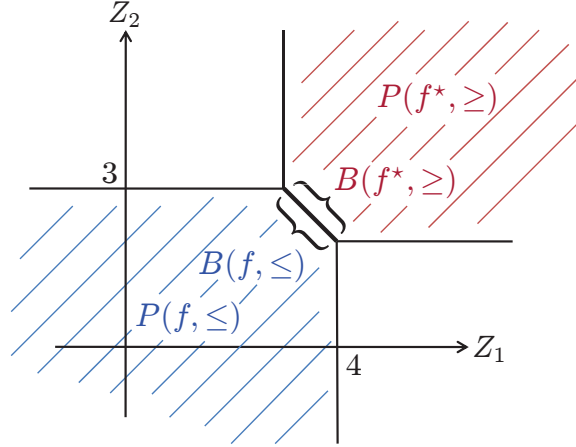


Fig. 3. Equivalence between $B(f, \leq)$ and $B(f^*, \geq)$ illustrated for the function f provided in Example 1.

problems are equivalent if they have the same optimal value and the same set of optimizers.

Lemma 2. *If $B(f) \neq \emptyset$, then the following optimization problems are equivalent*

$$\max_{\mathbf{Z}} \sum_{i=1}^m Z_i, \quad s.t. \quad \mathbf{Z} \in P(f, \leq). \quad (15)$$

$$\min_{\mathbf{R}} \sum_{i=1}^m R_i, \quad s.t. \quad \mathbf{R} \in P(f^*, \geq). \quad (16)$$

Lemma 2 can be easily proved from the following argument provided in [13]. Since $B(f, \leq) \neq \emptyset$, there exists a vector \mathbf{Z} such that $Z(\mathcal{M}) = f(\mathcal{M}) = f^*(\mathcal{M})$. Moreover, $\mathbf{Z} \in B(f, \leq) = B(f^*, \geq)$. Hence, \mathbf{Z} is a maximizer of the problem (15) and a minimizer of the problem (16).

Next, we define the class of *submodular functions* for which the maximization problem (15) has analytical solution.

Definition 4 (Submodularity). A set function f defined on the power set of \mathcal{M} , $f : 2^{\mathcal{M}} \rightarrow \mathbb{R}$, where $f(\emptyset) = 0$, is called *submodular* if

$$f(\mathcal{S}) + f(\mathcal{T}) \geq f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T}), \quad \forall \mathcal{S}, \mathcal{T} \subseteq \mathcal{M}. \quad (17)$$

Remark 1. When f is submodular, then $B(f, \leq) \neq \emptyset$.

For a more general version of the problem (15)

$$\max_{\mathbf{Z}} \sum_{i=1}^m \alpha_i Z_i, \quad \text{s.t. } \mathbf{Z} \in P(f, \leq), \quad (18)$$

where $\alpha_i \geq 0$, for $i = 1, \dots, m$, and f is submodular, an analytical solution can be obtained using Edmond's algorithm.

Theorem 1 (Edmond's greedy algorithm [14]). *When f is submodular, the maximization problem (18) given by $\max_{\mathbf{Z}} \sum_{i=1}^m \alpha_i Z_i$, s.t. $\mathbf{Z} \in P(f)$, can be solved analytically as follows.*

$$Z_{j(i)} = f(\mathcal{A}_i) - f(\mathcal{A}_{i-1}), \quad i = 1, \dots, m,$$

where $j(1), j(2), \dots, j(m)$ is an ordering of $\{1, 2, \dots, m\}$ such that $\alpha_{j(1)} \geq \alpha_{j(2)} \geq \dots \geq \alpha_{j(m)}$, and

$$\mathcal{A}_i = \emptyset, \quad i = 1,$$

$$\mathcal{A}_i = \{j(1), j(2), \dots, j(i)\}, \quad i = 2, 3, \dots, m.$$

The following statement directly follows from Remark 1.

Remark 2. When f is submodular, a maximizer \mathbf{Z} of the optimization problem (18) satisfies $\sum_{i=1}^m Z_i = f(\mathcal{M})$.

Example 2. In this example we illustrate Edmond's greedy algorithm by considering the set function f from Example 1 and the optimization problem

$$\max_{\mathbf{Z}} 5Z_1 + Z_2, \quad \text{s.t. } \mathbf{Z} \in P(f, \leq), \quad (19)$$

where $\mathbf{Z} = (Z_1, Z_2)$. Since $\alpha_1 = 5 > \alpha_2 = 1$, we set 1, 2 to be the ordering of $\{1, 2\}$, i.e., $j(1) = 1$ and $j(2) = 2$. Then, by applying Edmond's algorithm we obtain $Z_1 = 4$, $Z_2 = 2$ to be the maximizer of the problem (19).

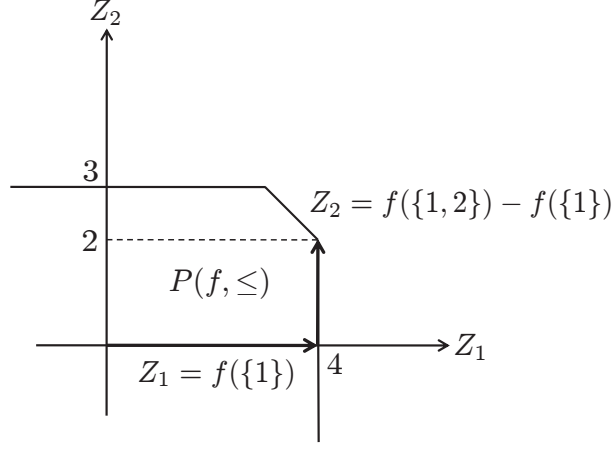


Fig. 4. Edmond's algorithm applied to the optimization problem (19). Since $\alpha_1 > \alpha_2$, the optimal ordering of $\{1, 2\}$ is 1, 2.

Edmond's algorithm is illustrated in Figure 4 for the case $\mathcal{M} = \{1, 2\}$. Notice that each iteration of the algorithm reaches a boundary of the polyhedron $P(f, \leq)$ until it finally reaches the vertex of the base polyhedron $B(f, \leq)$.

In [15], it was shown that the following optimization problem can also be solved using Edmond's greedy algorithm.

Corollary 1. When f is submodular, then the optimization problem

$$\min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i, \quad \text{s.t. } \mathbf{R} \in B(f, \leq), \quad (20)$$

can be solved by using Edmond's algorithm where $j(1), j(2), \dots, j(m)$ is an ordering of \mathcal{M} such that $\alpha_{j(1)} \leq \alpha_{j(2)} \leq \dots \leq \alpha_{j(m)}$.

Next, we introduce the class of *intersecting submodular* functions which is instrumental to solving our communication for omniscience problem.

Definition 5 (Intersecting Submodularity). A function f defined on the power set of \mathcal{M} , $f : 2^{\mathcal{M}} \rightarrow \mathbb{R}$ is called an *intersecting submodular* if

$$f(\mathcal{S}) + f(\mathcal{T}) \geq f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T}), \quad \forall \mathcal{S}, \mathcal{T} \text{ s.t. } \mathcal{S} \cap \mathcal{T} \neq \emptyset. \quad (21)$$

Notice that every submodular function is also intersecting submodular. However, in general, Edmond's algorithm cannot be directly applied to solve the maximization problem (18) over the polyhedron of an intersecting submodular function.

In [13] it is shown that for every intersecting submodular function there exists a submodular function such that both functions have the same polyhedron. This is formally stated in the following theorem.

Theorem 2 (Dilworth truncation). *For an intersecting submodular function $f : 2^{\mathcal{M}} \rightarrow \mathbb{R}$ with $f(\emptyset) = 0$, there exists a submodular function $g : 2^{\mathcal{M}} \rightarrow \mathbb{R}$ such that $g(\emptyset) = 0$ and $P(g, \leq) = P(f, \leq)$. The function g can be expressed as*

$$g(\mathcal{S}) = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{V} \in \mathcal{P}} f(\mathcal{V}) : \mathcal{P} \text{ is a partition of } \mathcal{S} \right\}. \quad (22)$$

The function g is called the Dilworth truncation of f .

Example 3. Let $\mathcal{M} = \{1, 2\}$, and $f(\{1\}) = 4$, $f(\{2\}) = 3$, $f(\{1, 2\}) = 8$. It is easy to verify that the function f is intersecting submodular, but not fully submodular since $f(\{1\}) + f(\{2\}) < f(\{1, 2\})$. Applying Dilworth truncation to the function f , we obtain g , where $g(\{1\}) = 4$, $g(\{2\}) = 3$, $g(\{1, 2\}) = 7$. Moreover, it can be checked that $P(g, \leq) = P(f, \leq)$.

If the Dilworth truncation g of the intersecting submodular function f is given, the optimization problem (18) can be efficiently solved using Edmond's greedy algorithm. However, finding the value of function g , even for a single set $\mathcal{S} \subseteq M$, involves a minimization over a set of exponential size (see (22)). This can be overcome using the facts that $P(g, \leq) = P(f, \leq)$, and that the maximizer of the problem (18) belongs to the base polyhedron $B(g, \leq)$ by Remark 1. The result is a modified version of Edmond's algorithm that can solve the optimization problem in polynomial time.

Lemma 3 (Modified Edmond's algorithm, [13], [16]). *When f is intersecting submodular, the maximization problem (18) given by $\max_{\mathbf{Z}} \sum_{i=1}^m Z_i$, s.t. $\mathbf{Z} \in P(f)$, can be solved as follows.*

Algorithm 1 Modified Edmond's Algorithm

- 1: Set $j(1), j(2), \dots, j(m)$ to be an ordering of $\{1, 2, \dots, m\}$ such that $\alpha_{j(1)} \geq \alpha_{j(2)} \geq \dots \geq \alpha_{j(m)}$
 - 2: Initialize $\mathbf{Z} = \mathbf{0}$.
 - 3: **for** $i = 1$ to m **do**
 - 4: $Z_{j(i)} = \min_{\mathcal{S}} \{f(\mathcal{S}) - Z(\mathcal{S}) : j(i) \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{A}_i\}$.
 - 5: **end for**
-

The following statement directly follows from Theorem 2 and Remark 2.

Remark 4. The submodular function minimization routine can be done in polynomial time. The best known algorithm to our knowledge is proposed by Orlin in [7], and has complexity $\mathcal{O}(m^5 \cdot \gamma + m^6)$, where γ is complexity of computing the submodular function.

IV. COMMUNICATION FOR OMNISCIENCE RATES

In this section we propose an efficient algorithm for computing a rate tuple which belongs to $\mathcal{R}(\underline{\alpha})$, *i.e.*, an optimal rate tuple w.r.t. the optimization problem

$$\min_{\mathbf{R}} \sum_{i=1}^m R_i, \quad \text{s.t. } \mathbf{R} \in \mathcal{R}. \quad (23)$$

We start with the special case when $\underline{\alpha} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$, henceforth denoted as $\underline{\alpha} = \mathbf{1}$. This instance represents a key building block for solving the problem for general cost vector $\underline{\alpha}$. We begin by observing that the rate region defined in (3) can be represented as a polyhedron of some set function, say f^* , to be defined later. In this section we solve $\text{LP}_1(\mathbf{1})$ by considering the dual set function f of f^* , and solving the corresponding dual optimization problem. We show that it is possible to construct a function f^* defining the rate region \mathcal{R} such that its dual function f is *intersecting submodular*. Therefore, the underlying optimization problem can be solved in polynomial time using the modified Edmond's algorithm. Therefore, the optimization problem $\text{LP}_1(\mathbf{1})$ can be stated as follows

$$\min_{\mathbf{R}} \sum_{i=1}^m R_i, \quad \text{s.t. } \mathbf{R} \in P(f^*, \geq), \quad (24)$$

where $P(f^*)$ is a polyhedron of a set function f^* such that $P(f^*, \geq) = \mathcal{R}$. To that end, we can choose

$$f^*(S) = H(X_S | X_{S^c}), \quad \forall S \subset \mathcal{M}. \quad (25)$$

Notice that the function f^* is not completely defined in (25) because the value of $f^*(\mathcal{M})$ is missing. Therefore, we need to assign $f^*(\mathcal{M})$ such that $P(f^*, \geq) = \mathcal{R}$ and $B(f^*, \geq) \neq \emptyset$. The second condition ensures equivalence between the optimization problem (24) and the corresponding dual problem (see Lemma 2). It is not hard to see that taking $f^*(\mathcal{M}) = R_{CO}(\mathbf{1})$ satisfies all the conditions above. Thus, we have

$$f^*(S) = \begin{cases} H(X_S | X_{S^c}) & \text{if } S \subset \mathcal{M}, \\ R_{CO}(\mathbf{1}) & \text{if } S = \mathcal{M}. \end{cases} \quad (26)$$

Of course $R_{CO}(\mathbf{1})$ is not known a priori, but this issue will be addressed later. According to Definition 3, the dual set function f of f^* has the following form

$$f(\mathcal{S}) = \begin{cases} R_{CO}(\mathbf{1}) - H(X_{\mathcal{S}^c}|X_{\mathcal{S}}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \quad (27)$$

Using the duality result in Lemma 2, it follows that the optimization problem (24) is equivalent to

$$\max_{\mathbf{Z}} \sum_{i=1}^m Z_i, \quad \text{s.t. } \mathbf{Z} \in P(f, \leq). \quad (28)$$

To avoid cumbersome expressions, hereafter we use $P(f)$ and $B(f)$ to denote $P(f, \leq)$ and $B(f, \leq)$, respectively. Hence, the optimal value of the optimization problem (28) is $R_{CO}(\mathbf{1})$. However, the value of $R_{CO}(\mathbf{1})$ is not known a priori. To that end, let us replace $R_{CO}(\mathbf{1})$ in (27) with a variable β , and construct a two-argument function $f(\mathcal{S}, \beta)$ as follows.

$$f(\mathcal{S}, \beta) \triangleq \begin{cases} \beta - H(X_{\mathcal{S}^c}|X_{\mathcal{S}}) & \text{if } \emptyset \neq \mathcal{S} \subseteq \mathcal{M}, \\ 0 & \text{if } \mathcal{S} = \emptyset. \end{cases} \quad (29)$$

Lemma 4. *Function $f(\mathcal{S}, \beta)$ defined in (29) is intersecting submodular. When $\beta \geq H(X_{\mathcal{M}})$, the function $f(\mathcal{S}, \beta)$ is submodular.*

Proof of Lemma 4 is provided in Appendix B. Considering the optimization problem

$$\max_{\mathbf{Z}} \sum_{i=1}^m Z_i, \quad \text{s.t. } \mathbf{Z} \in P(f, \beta), \quad (30)$$

as a function of β , the goal is to identify its characteristics at the point $\beta = R_{CO}(\mathbf{1})$. Hereafter, we refer to the optimization problem (30) as $LP_2(\beta)$.

Theorem 4. *The optimal value $R_{CO}(\mathbf{1})$ can be obtained as follows*

$$R_{CO}(\mathbf{1}) = \min \beta \text{ such that } \beta \text{ is the optimal value of } LP_2(\beta). \quad (31)$$

Proof: We prove this theorem by contradiction. First, notice that $\beta = R_{CO}(\mathbf{1})$ is a feasible solution for the optimization problem (31). Next, let us assume that for some $\beta' < R_{CO}(\mathbf{1})$ there exists a vector \mathbf{Z} that is a maximizer of the problem $LP_2(\beta')$ such that $Z(\mathcal{M}) = \beta' = f(\mathcal{M}, \beta')$. Since $\mathbf{Z} \in P(f, \beta')$ it must satisfy the following set of inequalities

$$Z(\mathcal{S}) \leq \beta' - H(X_{\mathcal{S}^c}|X_{\mathcal{S}}), \quad \forall \emptyset \neq \mathcal{S} \subseteq \mathcal{M}. \quad (32)$$

Since $\beta' = Z(\mathcal{M})$, and $Z(\mathcal{S}^c) = Z(\mathcal{M}) - Z(\mathcal{S})$, we can write (32) as

$$Z(\mathcal{S}^c) \geq H(X_{\mathcal{S}^c}|X_{\mathcal{S}}), \quad \forall \emptyset \neq \mathcal{S} \subset \mathcal{M}. \quad (33)$$

Therefore, $\mathbf{Z} \in \mathcal{R}$ is a feasible rate tuple w.r.t. the optimization problem $\text{LP}_1(\mathbf{1})$ and, hence, it must hold that $\beta' \geq R_{CO}(\mathbf{1})$. This is in contradiction with our previous statement that $\beta' < R_{CO}(\mathbf{1})$. ■

Since $R_{CO}(\mathbf{1})$ can be trivially upper bounded by $H(X_{\mathcal{M}})$ and lower bounded by 0, we can restrict the search space in (31) to $0 \leq \beta \leq H(X_{\mathcal{M}})$.

Function $f(\mathcal{S}, \beta)$ is intersecting submodular for the case of interest when $0 \leq \beta \leq H(X_{\mathcal{M}})$. As noted in Theorem 2, for the intersecting submodular function $f(\mathcal{S}, \beta)$, there exists a submodular function, here denoted by Dilworth truncation $g(\mathcal{S}, \beta)$, such that $P(f, \beta) = P(g, \beta)$.

$$g(\mathcal{S}, \beta) = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{V} \in \mathcal{P}} f(\mathcal{V}, \beta) : \mathcal{P} \text{ is a partition of } \mathcal{S} \right\}. \quad (34)$$

Definition 6. Let $\mathcal{P}(\beta)$ denote an optimal partitioning of the set \mathcal{M} according to (34) for the given β .

From Remark 3 it follows that $g(\mathcal{M}, \beta)$ is the optimal value of the optimization problem $\text{LP}_2(\beta)$ for any given β . Hence, it can be obtained in polynomial time by applying the modified Edmond's algorithm to the set function $f(\mathcal{S}, \beta)$. Moreover, the corresponding optimal partition $\mathcal{P}(\beta)$ can be efficiently obtained by adding two additional steps to the modified Edmond's algorithm as shown in [16] and [13] (see Algorithm 3 in Appendix D).

From Theorem 4, it follows that the optimal omniscience rate $R_{CO}(\mathbf{1})$ can be calculated as follows:

$$R_{CO}(\mathbf{1}) = \min_{0 \leq \beta \leq H(X_{\mathcal{M}})} \beta, \quad \text{s.t. } g(\mathcal{M}, \beta) = \beta. \quad (35)$$

Notice that $g(\mathcal{M}, \beta) = f(\mathcal{M}, \beta) = \beta$ whenever the optimal partitioning of the set \mathcal{M} according to (34) is of cardinality 1, i.e., $\mathcal{P}(\beta) = \{\{\mathcal{M}\}\}$.

In the further text we show how to solve the optimization problem (35) with at most m calls of the modified Edmond's algorithm, which makes the complexity of the entire algorithm polynomial in m . From (34) it follows that for every β , the function $g(\mathcal{M}, \beta)$ can be represented as

$$g(\mathcal{M}, \beta) = |\mathcal{P}(\beta)|\beta - \sum_{\mathcal{S} \in \mathcal{P}(\beta)} H(X_{\mathcal{S}^c}|X_{\mathcal{S}}). \quad (36)$$

Therefore, $g(\mathcal{M}, \beta)$ is piecewise linear in β .

Lemma 5. *Function $g(\mathcal{M}, \beta)$ has the following properties*

- 1) *It has at most m linear segments.*

2) It has non-increasing slope, i.e., $g(\mathcal{M}, \beta)$ is a concave function.

3) The last linear segment is of slope 1.

Moreover, $\beta = R_{CO}(\mathbf{1})$ represents a breakpoint in $g(\mathcal{M}, \beta)$ between the linear segment with slope 1 and consecutive linear segment with the larger slope.

The proof of Lemma 5 is provided in Appendix C. From (36) it follows that the slope of the function $g(\mathcal{M}, \beta)$ is equal to the cardinality of the optimal partition $\mathcal{P}(\beta)$. Since there are at most m linear segments in $g(\mathcal{M}, \beta)$, we can solve for the breakpoint of interest according to Lemma 5 in polynomial time by performing a binary search. We explain this procedure on a simple case described in Figure 6. From Lemma 5 we have that $\beta = R_{CO}(\mathbf{1})$ is a breakpoint of $g(\mathcal{M}, \beta)$ between the linear segment

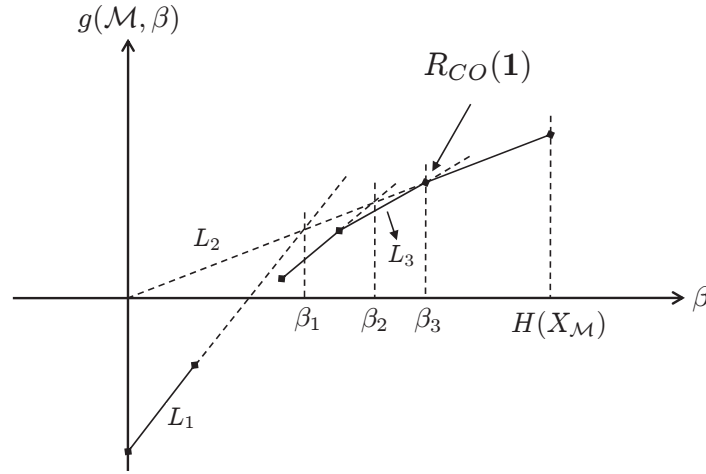


Fig. 6. Optimal $R_{CO}(\mathbf{1})$ can be obtained by intersecting linear segments. First, we intersect the line L_1 which corresponds to $\beta = 0$, with the 45-degree line L_2 . The intersecting point β_1 belongs to the linear segment with slope greater than 1. Then, intersecting the segment L_3 to which β_1 belongs to with the 45-degree line L_2 , we obtain β_2 , and finally β_3 after one more intersection. Since the linear segment at β_3 has slope 1, we conclude that $\beta_3 = R_{CO}(\mathbf{1})$.

with slope 1 and consecutive linear segment with the larger slope. Moreover, for every β one can obtain a value of $g(\mathcal{M}, \beta)$ and the corresponding optimal partition $\mathcal{P}(\beta)$ w.r.t. (34) in polynomial time using Algorithm 3 in Appendix D. Due to concavity of $g(\mathcal{M}, \beta)$, the following algorithm will converge to the breakpoint $\beta = R_{CO}(\mathbf{1})$ in at most m iterations.

Since $R_{CO}(\mathbf{1}) \geq 0$, we start by, first, intersecting the line L_1 which belongs to the linear segment when $\beta = 0$ and the 45-degree line L_2 which corresponds to the last (rightmost) linear segment. Slope of the line L_1 as well as its value can be obtained in polynomial time by applying Algorithm 3 for $\beta = 0$.

Since the function $g(\mathcal{M}, \beta)$ is piecewise linear and concave, the point of intersection β_1 must belong to the linear segment with slope smaller than $|\mathcal{P}(0)|$, i.e., $|\mathcal{P}(\beta_1)| < |\mathcal{P}(0)|$. β_1 can be obtained by equating β with $\sum_{S \in \mathcal{P}(0)} \beta - H(X_{S^c}|X_S)$. Hence,

$$\beta_1 = \frac{\sum_{S \in \mathcal{P}(0)} H(X_{S^c}|X_S)}{|\mathcal{P}(0)| - 1}. \quad (37)$$

Next, by applying Algorithm 3 for $\beta = \beta_1$, we get $(g(\mathcal{M}, \beta_1), \mathcal{P}(\beta_1))$. Since $|\mathcal{P}(\beta_1)| > 1$ (see Figure 6), we have not reached the breakpoint of interest yet, because $R_{CO}(\mathbf{1})$ belongs to the linear segment of slope 1. Thus, we proceed by intersecting the line L_3 which belongs to the linear segment when $\beta = \beta_1$ with the the 45-degree line L_2 . Like in the previous case, we obtain $\beta_2 = \frac{\sum_{S \in \mathcal{P}(\beta_1)} H(X_{S^c}|X_S)}{|\mathcal{P}(\beta_1)| - 1}$. Since $|\mathcal{P}(\beta_2)| > 1$, we need to perform one more intersection to obtain β_3 for which $|\mathcal{P}(\beta_3)| = 1$. Hence, $\beta_3 = R_{CO}(\mathbf{1})$. For an arbitrary $g(\mathcal{M}, \beta)$, the binary search algorithm can be constructed as follows.

Algorithm 2 Achieving a rate tuple from the region $\mathcal{R}(\mathbf{1})$

- 1: Initialize $\beta = 0$.
 - 2: **while** $|\mathcal{P}(\beta)| > 1$ **do**
 - 3: $\beta = \frac{\sum_{S \in \mathcal{P}(\beta)} H(X_{S^c}|X_S)}{|\mathcal{P}(\beta)| - 1}$, where $\mathcal{P}(\beta)$ is obtained from Algorithm 3.
 - 4: **end while**
 - 5: $\beta = R_{CO}(\mathbf{1})$.
-

It is not hard to see that Algorithm 2 executes at most m iterations, since with each iteration the intersection point moves to the right to some other linear segment until it hits $R_{CO}(\mathbf{1})$ (see Figure 6).

Therefore, Algorithm 2 calls Algorithm 3 at most m times. Since the complexity of Algorithm 3 is $\mathcal{O}(m \cdot SFM(m))$ (see Appendix D), the total complexity of obtaining a rate tuple that belongs to $\mathcal{R}(\mathbf{1})$ through Algorithm 2 is $\mathcal{O}(m^2 \cdot SFM(m))$.

V. ACHIEVING A RATE TUPLE THAT BELONGS TO $\mathcal{R}(\underline{\alpha})$

In this section we investigate the problem of computing a rate tuple that belongs to $\mathcal{R}(\underline{\alpha})$, where $0 \leq \alpha_i < \infty$, $i = 1, 2, \dots, m$. We propose an algorithm of polynomial complexity that is based on the results we derived for the $\mathcal{R}(\mathbf{1})$ case.

Let us start with restating the optimization problem $LP_1(\underline{\alpha})$ in the following way.

$$\min_{\beta} \min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i \quad \text{s.t.} \quad R(\mathcal{M}) = \beta, \quad R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subset \mathcal{M} \quad (38)$$

where $\beta \geq R_{CO}(\mathbf{1})$. Hereafter we denote optimization problem (38) by $\text{LP}_3(\alpha)$. This interpretation of the problem $\text{LP}_1(\underline{\alpha})$ corresponds to finding its optimal value by searching over all achievable sum rates $R(\mathcal{M})$. Let us focus on the second term in optimization (38).

$$\min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i \quad \text{s.t.} \quad R(\mathcal{M}) = \beta, \quad R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subset \mathcal{M}. \quad (39)$$

Observe that the rate region in (39) constitutes a base polyhedron $B(f^*, \beta, \geq)$, where

$$f^*(\mathcal{S}, \beta) = \begin{cases} H(X_{\mathcal{S}}|X_{\mathcal{S}^c}) & \text{if } \mathcal{S} \subset \mathcal{M}, \\ \beta & \text{if } \mathcal{S} = \mathcal{M}. \end{cases} \quad (40)$$

Since $\beta \geq R_{CO}(\mathbf{1})$ we have that $B(f^*, \beta, \geq) \neq \emptyset$. From Lemma 1 it follows that $B(f^*, \beta, \geq) = B(f, \beta)$, where $f(\mathcal{S}, \beta)$, defined in (29), is a dual set function of $f^*(\mathcal{S}, \beta)$. Hence, the optimization problem (39) is equivalent to

$$\min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i \quad \text{s.t.} \quad \mathbf{R} \in B(f, \beta). \quad (41)$$

In Corollary 1 we implied that for any fixed $\beta \geq R_{CO}(\mathbf{1})$ the optimization problem (41) can be solved using Edmond's algorithm, with $j(1), j(2), \dots, j(m)$ being the ordering of \mathcal{M} such that $\alpha_{j(1)} \leq \alpha_{j(2)} \leq \dots \leq \alpha_{j(m)}$. However, since the function $f(\mathcal{S}, \beta)$ is intersecting submodular, it is necessary to apply the modified version of Edmond's algorithm provided in Lemma 3 to obtain an optimal rate tuple w.r.t. (41).

Let $h(\beta)$ denote the optimal value of the optimization problem defined in (41)

$$h(\beta) = \min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i \quad \text{s.t.} \quad \mathbf{R} \in B(f, \beta). \quad (42)$$

To that end, we can state problem $\text{LP}_3(\underline{\alpha})$ as

$$\min_{\beta} h(\beta), \quad \text{s.t.} \quad \beta \geq R_{CO}(\mathbf{1}). \quad (43)$$

With every $\beta \geq R_{CO}(\mathbf{1})$ we associate an optimal rate vector \mathbf{R} w.r.t. optimization problem (42). Next, we show some basic properties of the function $h(\beta)$.

Lemma 6. *Function $h(\beta)$ defined in (42) is continuous and convex when $\beta \geq R_{CO}(\mathbf{1})$.*

Proof of Lemma 6 is provided in Appendix E.

A. Gradient Descent Method

From Lemma 6 it immediately follows that we can apply a gradient descent algorithm to minimize the function $h(\beta)$. However, in order to do that, at every point β , we need to know the value of $h(\beta)$ as well

as its derivative. As mentioned above, an optimal rate tuple that corresponds to the function $h(\beta)$ can be obtained by applying the modified Edmond's algorithm to the problem (41). From Lemma 3 it follows that the optimal rate vector with respect to the optimization problem (41), has the following form.

$$R_i = b_i \cdot \beta + c_i, \quad \forall i \in \mathcal{M}, \quad (44)$$

where $b_i \in \mathbb{Z}$, and c_i is a constant which corresponds to a summation of some conditional entropy terms. Moreover, it follows that the coefficients (b_i, c_i) , $i = 1, 2, \dots, m$, depend only on the value of β (they do not depend on the weight vector $\underline{\alpha}$).

Lemma 7. *Function $h(\beta)$ is piecewise linear in β . For a fixed $\beta \geq R_{CO}(\mathbf{1})$ the values of $h(\beta)$ and $\frac{dh(\beta)}{d\beta}$ can be obtained in $\mathcal{O}(m \cdot SFM(m))$ time by applying the modified Edmond's algorithm to the ordering of \mathcal{M} specified in Corollary 1. Derivative of $h(\beta)$ can be calculated by expressing the optimal rates R_i , $i \in \mathcal{M}$, as $R_i = b_i \cdot \beta + c_i$ in each iteration of the modified Edmond's algorithm. Then,*

$$\frac{dh(\beta)}{d\beta} = \sum_{i=1}^m \alpha_i \cdot b_i. \quad (45)$$

To make the gradient descent algorithm more efficient, it is useful to make a search space as tight as possible. So far, we showed that the minimizer of the problem $LP_3(\underline{\alpha})$ belongs to the region $[R_{CO}(\mathbf{1}), \infty)$. Combining the results of Lemma 6 and Lemma 7, we have the following bound.

Lemma 8. *Let β^* be the minimizer of the optimization problem $LP_3(\underline{\alpha})$. Then,*

$$R_{CO}(\mathbf{1}) \leq \beta^* \leq H(X_{\mathcal{M}}). \quad (46)$$

Proof: Note that the function $f(\mathcal{S}, \beta)$ is submodular when $\beta = H(X_{\mathcal{M}})$ (see Lemma 4). Optimization problem (39) for $\beta = H(X_{\mathcal{M}})$, can be solved by applying Edmond's algorithm (see Theorem 1) to the optimization problem (41). It is easy to verify that the optimal rates have the following form:

$$\begin{aligned} R_{j(1)} &= \beta + c_{j(1)}, \\ R_{j(i)} &= c_{j(i)}, \quad i \in \{2, 3, \dots, m\}. \end{aligned}$$

Hence,

$$h(\beta = H(X_{\mathcal{M}})) = \alpha_{j(1)}\beta + \sum_{i=1}^m \alpha_{j(i)}c_{j(i)}.$$

Since $\alpha_{j(1)} \geq 0$, and function $h(\beta)$ is convex, it immediately follows that $\beta^* \leq H(X_{\mathcal{M}})$. ■

Since the function $h(\beta)$ is continuous and differentiable, we can find its minimum, and therefore solve the optimization problem $LP_3(\underline{\alpha})$, by applying a gradient descent algorithm. However, in general case,

we can only reach the optimal point up to some precession ε . In order to be at most ε away from the optimal solution, the gradient descent method executes approximately $\mathcal{O}(\log \frac{1}{\varepsilon})$ iterations [17]. Therefore, the total complexity of obtaining a rate tuple with a sum rate that is at most ε away from the optimal one is $\mathcal{O}(m^2 \cdot SFM(m) + \log \frac{H(X_{\mathcal{M}})}{\varepsilon} \cdot m \cdot SFM(m))$, where the first term corresponds to the complexity of finding $R_{CO}(1)$.

Before we go any further, let us briefly analyze a solution to the optimization problem $LP_1(\underline{\alpha})$. We can think of it as a minimal value C for which the plane $C - \sum_{i=1}^m \alpha_i R_i$ intersects the rate region \mathcal{R} defined in (3). It is not hard to conclude that the point of intersection is one of the “vertices” of the region \mathcal{R} , i.e., it is completely defined by the collection of sets $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m\}$ such that

$$R(\mathcal{S}_i) = H(X_{\mathcal{S}_i} | X_{\mathcal{S}_i^c}), \quad i \in \{1, 2, \dots, m\}.$$

The following theorem will be very useful in Section VI when we explore the finite linear source model. It represents a key building block for bounding the total number of breakpoints in $h(\beta)$.

Theorem 5. *For every breakpoint of the function $h(\beta)$, the corresponding rate vector \mathbf{R} that minimizes (42) is a vertex of the rate region \mathcal{R} .*

Proof: Due to the equivalence between the problems $LP_1(\underline{\alpha})$ and $LP_3(\underline{\alpha})$ it follows that for every $\underline{\alpha}$, the rate tuple \mathbf{R} which corresponds to the minimizer of the function $h(\beta)$, is a vertex of the rate region \mathcal{R} . For a given cost vector $\underline{\alpha}$, we prove this theorem by modifying $\underline{\alpha}$ such that each breakpoint in $h(\beta)$ can become the minimizer of the function h that corresponds to the modified vector $\underline{\alpha}$.

To that end, let us consider an example of $h(\beta)$ shown in Figure 7. Each linear segment of $h(\beta)$ is described by a pair of vector $(\mathbf{b}^{(i)}, \mathbf{c}^{(i)})$, $i = 1, 2, 3, 4$, as in (44). Function $h(\beta)$ is minimized when $\beta = \beta_3$.

First, we show how to modify $\underline{\alpha}$ so that the breakpoint β_2 becomes the minimizer of $LP_3(\underline{\alpha})$. From (44), we have that the slopes of the segments $[\beta_1, \beta_2]$ and $[\beta_2, \beta_3]$ are such that $\sum_{i=1}^m \alpha_i b_i^{(1)} < 0$, $\sum_{i=1}^m \alpha_i b_i^{(2)} < 0$. Since $h(\beta)$ is convex, it also holds that

$$\sum_{i=1}^m \alpha_i b_i^{(1)} < \sum_{i=1}^m \alpha_i b_i^{(2)}. \quad (47)$$

Observe that for every $\beta \geq R_{CO}(1)$, the rate tuple that corresponds to $h(\beta)$ is such that $R(\mathcal{M}) = \beta$. Hence, for each linear segment it holds that $\sum_{j=1}^m b_j^{(i)} = 1$, $i = 1, 2, 3, 4$. Let

$$\alpha'_i = \alpha_i + \Delta\alpha, \quad i = 1, 2, \dots, m, \quad (48)$$

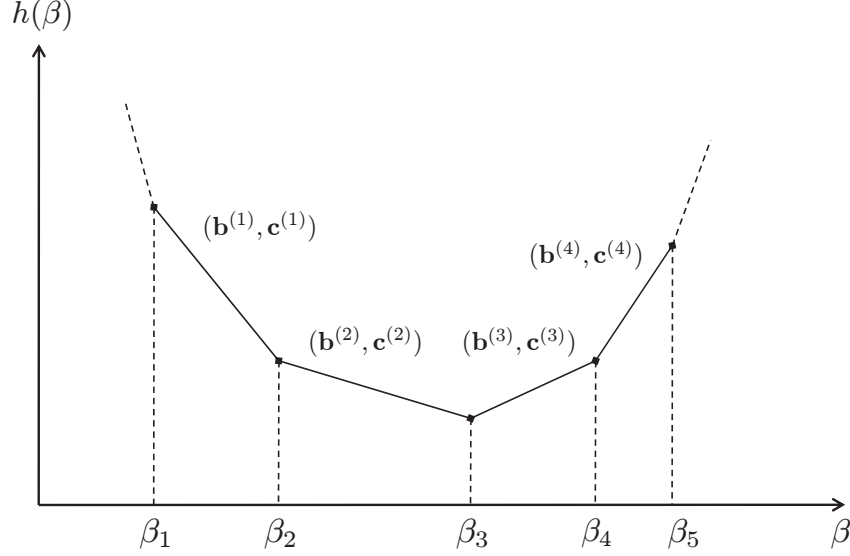


Fig. 7. Function $h(\beta)$ is a piecewise linear in β . For the purpose of proving Theorem 5, we consider 4 linear segments, and show that each breakpoint can become the minimizer of a different optimization problem.

where $\Delta\alpha > 0$ is a constant. For the weight vector $\underline{\alpha}'$ constructed in (48), the segments $[\beta_1, \beta_2]$ and $[\beta_2, \beta_3]$ have slopes

$$\sum_{i=1}^m b_i^{(j)}(\alpha_i + \Delta\alpha) = \sum_{i=1}^m b_i^{(j)}\alpha_i + \Delta\alpha, \quad j = 1, 2.$$

Therefore, we can pick $\Delta\alpha$ such that the linear segment $[\beta_1, \beta_2]$ has negative slope, while the linear segment $[\beta_2, \beta_3]$ has positive slope. One possible choice is

$$\Delta\alpha = -\sum_{i=1}^m \alpha_i b_i^{(2)} + \epsilon, \quad (49)$$

where ϵ is a small positive constant. Note that due to (47) the linear segment $[\beta_1, \beta_2]$ still has negative slope. Similarly, we can move a minimizer of $\text{LP}_3(\underline{\alpha})$ from β_3 to β_4 , by modifying $\underline{\alpha}$ as follows

$$\alpha'_i = \alpha_i - \Delta\alpha, \quad i = 1, 2, \dots, m, \quad (50)$$

where $\Delta\alpha = \sum_{i=1}^m \alpha_i b_i^{(3)} + \epsilon$. In this case, linear segments $[\beta_3, \beta_4]$ and $[\beta_4, \beta_5]$ have the slopes $\sum_{i=1}^m \alpha'_i b_i^{(3)} < 0$ and $\sum_{i=1}^m \alpha'_i b_i^{(4)} > 0$, which makes $\beta = \beta_4$ the minimizer of $\text{LP}_3(\underline{\alpha}')$. Therefore, we showed how to modify the cost vector $\underline{\alpha}$, so that the minimizer of $h(\beta)$ “jumps” to the consecutive breakpoints of $h(\beta)$. Repeating this procedure multiple times, one can modify $\underline{\alpha}$ so that any breakpoint becomes the minimizer of $h(\beta)$. ■

VI. DATA EXCHANGE PROBLEM WITH LINEAR CORRELATIONS

In this section we propose a polynomial time algorithm for achieving a rate tuple that belongs to the region $\mathcal{R}(\underline{\alpha})$ in the data exchange problem. In Section II we defined a linear model where each user $i \in \mathcal{M}$ observes a collection of the linear equations in \mathbb{F}_{q^n} ,

$$X_i = \mathbf{A}_i \mathbf{W}, \quad i \in \mathcal{M}, \quad (51)$$

where $\mathbf{A}_i \in \mathbb{F}_q^{\ell_i \times N}$ is a fixed matrix and $\mathbf{W} \in \mathbb{F}_{q^n}^N$ is a vector of data packets. Since all the algebraic operations are performed over the base field \mathbb{F}_q , the linear model (51) is equivalent to the scenario where each user observes n memoryless instances of the finite linear process (51) where \mathbf{W} is a uniform vector over \mathbb{F}_q^N . Hereafter, we will use the entropy of the observations and the rank of the observation matrix interchangeably.

Theorem 6. *For the linear source model, any rate tuple \mathbf{R} that belongs to the rate region \mathcal{R}_{de} , defined in (9), can be achieved via linear network coding, i.e., in order to achieve omniscience it is sufficient for each user $i \in \mathcal{M}$ to transmit R_i properly chosen linear equations of the data packets he observes.*

Proof of Theorem 6 is provided in Appendix G. This result suggests that in an optimal communication scheme, each user transmits some integer number of symbols in \mathbb{F}_q . Hence, a rate tuple that belongs to $\mathcal{R}(\underline{\alpha})$ in the data exchange problem has to be some fractional number with the denominator n . To that end, we introduce a fractional rate constraint to the optimization problem $\text{LP}_1(\underline{\alpha})$ in order to obtain the optimal solution for the data exchange problem.

$$\min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i, \quad \text{s.t.} \quad R(\mathcal{S}) \geq H(X_{\mathcal{S}} | X_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subset \mathcal{M}, \quad (52)$$

where $n \cdot R_i \in \mathbb{Z}$, $\forall i \in \mathcal{M}$. Optimization problem (52) is an integer linear program, henceforth denoted by $\text{ILP}_n(\underline{\alpha})$. We use $\mathcal{R}_n(\underline{\alpha})$ to denote the rate region of all minimizers of the above ILP, and $R_{CO,n}(\underline{\alpha})$ to denote the minimal cost.

Notice that there is a certain gap between the “information-theoretic” optimal solution to the problem $\text{LP}_1(\underline{\alpha})$, and the “data exchange” optimal solution to the problem $\text{ILP}_n(\underline{\alpha})$. The reason is that the former solution assumes that the observation length tends to infinity, while in the data exchange setting we are dealing with the finite block lengths.

In this section we show how to efficiently solve $\text{ILP}_n(\underline{\alpha})$ by applying the optimization techniques we derived so far. Then, we propose a polynomial time code construction based on the matrix completion method over finite fields borrowed from the network coding literature [18].

To gain more insight into the coding scheme, let us start with the problem of finding a rate tuple that belongs to the region $\mathcal{R}_n(\mathbf{1})$.

A. Achieving a rate tuple from $\mathcal{R}_n(\mathbf{1})$

Let us consider the optimization problem $\text{ILP}_n(\mathbf{1})$. Observe that by applying the modified Edmond's algorithm for any $\beta \geq R_{CO}(\mathbf{1})$, we obtain a feasible rate tuple that corresponds to the rate region \mathcal{R}_{de} defined in (9). Moreover, by setting β to be a fractional number with the denominator n in the problem $\text{LP}_1(\beta)$, we also get all the optimal rates to be fractional numbers with the denominator n . Hence, an optimal rate tuple with respect to the optimization problem $\text{ILP}_n(\mathbf{1})$ can be obtained by applying the modified Edmond's algorithm for $\beta = \frac{\lfloor n \cdot R_{CO}(\mathbf{1}) \rfloor}{n} = R_{CO,n}(\mathbf{1})$. The next natural question is how far we are from the information-theoretic optimal solution, *i.e.*, when $n \rightarrow \infty$.

Claim 1. *The optimal sum rate w.r.t. $\text{ILP}_n(\mathbf{1})$ is at most $\frac{1}{n}$ symbols in \mathbb{F}_q away from $R_{CO}(\mathbf{1})$.*

$$R_{CO,n}(\mathbf{1}) - R_{CO}(\mathbf{1}) \leq \frac{1}{n}. \quad (53)$$

Example 5. Consider an example where 3 users observe the packets of length $n = 2$ over the field \mathbb{F}_q .

$$\begin{aligned} \mathbf{X}_1 &= [\mathbf{a} \quad \mathbf{b}], \\ \mathbf{X}_2 &= [\mathbf{a} \quad \mathbf{c}], \\ \mathbf{X}_3 &= [\mathbf{b} \quad \mathbf{c}], \end{aligned} \quad (54)$$

where $\mathbf{W} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}$ is a data packet vector in $\mathbb{F}_{q^2}^3$ such that $\mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} b_1 & b_2 \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$.

As pointed out above, we can think of this model as $n = 2$ repetitions of the finite linear process. Solving the problem $\text{ILP}_n(\mathbf{1})$ for this example, we obtain $R_1 = R_2 = R_3 = \frac{1}{2}$. Moreover, we also obtain the same rate allocation for the $\text{LP}_1(\mathbf{1})$, which suggests that in this case there is no gap in optimality between the finite and infinite observation length.

In Theorem 6 we showed that the network coding solution can achieve any rate tuple that belongs to \mathcal{R}_{de} , and hence, it also achieves any rate tuple from $\mathcal{R}_n(\mathbf{1})$. It is not hard to see that one possible solution for this example is: user 1 transmits $a_1 + b_2$, user 2 transmits $c_1 + a_2$, and user 3 transmits $b_1 + c_2$.

B. Code Construction

The next question that arises from this analysis is how to design the actual transmissions of each user? Starting from an optimal (integer) rate allocation, we construct the corresponding multicast network (see Figure 8). Then, using polynomial time algorithms for the multicast code construction [19], [18], we can solve for the actual transmissions of each user. We illustrate conversion of the data exchange problem to the multicast problem by considering the source model in Example 5. Then, the extension to an arbitrary linear source model is straightforward.

In this construction, notice that there are 4 different types of nodes. Conversion of our problem into the multicast problem assumes the existence of the super-node, here denoted by S , that possess all the packets. In the original problem, each user in the system plays the role of a transmitter and a receiver. To distinguish between these two states, we denote s_1, s_2 and s_3 to be the “sending” nodes, and r_1, r_2 and r_3 to be the “receiving” nodes which corresponds to the users 1, 2 and 3 in the original system, respectively.

Node S , therefore, feeds its information to the nodes s_1, s_2 and s_3 . Unlike the multicast problem, where any linear combination of the packets can be transmitted from node S to s_1, s_2 and s_3 , here the transmitted packets correspond to the observations of the users 1, 2 and 3, respectively. The second layer of the network is designed based on the optimal rates R_1, R_2 and R_3 . Since $n = 2$, each user gets to transmit 1 symbol in \mathbb{F}_q . It is clear that all the receiving users are getting two different types of information:

- 1) The side information that each user already has. In the multicast network this information is transmitted directly from node s_i to node $r_i, i = 1, 2, 3$.
- 2) The information that each node r_i receives from the other nodes $s_j, j \neq i$.

To model the second type of information, let us consider the nodes r_2 and r_3 .

Due to the broadcast nature of the channel, both r_2 and r_3 are receiving the same symbol in \mathbb{F}_q from node s_1 . Thus, it is necessary to introduce a dummy node t_1 to model this constraint. The capacities of the links $s_1 - t_1, t_1 - r_2$ and $t_1 - r_3$ are all equal to 1 symbol in \mathbb{F}_q . Note that this constraint ensures that the nodes r_2 and r_3 are obtaining the same 1 symbol from s_1 . The remaining edges are designed in a similar way.

Now, when we have a well-defined network, it is only left to figure out transmissions on all the edges. If we want to apply Jaggi’s algorithm [19], the first step is to determine disjoint paths from the super-node S to each receiver $r_1 - r_3$ using the Ford-Fulkerson algorithm [20]. While the solution to this

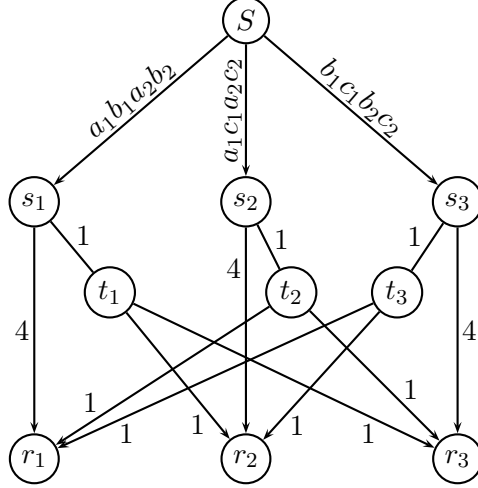


Fig. 8. Multicast network constructed from the source model and the optimal rate tuple $R_1 = R_2 = R_3 = \frac{1}{2}$ that belongs to $\mathcal{R}_2(1)$. Each user receives side information from “itself” (through the links $s_i - r_i$, $i = 1, 2, 3$) and the other users (through the links $t_i - r_j$, $i, j \in \{1, 2, 3\}$, $i \neq j$).

problem is easy in the case when each user observes only a subset of the packets (like in this example), it is not trivial to find disjoint paths which connect linearly independent sources to the receivers r_i (see Figure 8). For that reason we apply Harvey’s algorithm [18] which is based on matrix representation of the transmissions in the network [21], [22], and simultaneous matrix completion problem over finite fields.

In [21], the authors derived the transfer matrix $\mathbf{M}(r_i)$ from the super-node S to any receiver r_i , $i = 1, 2, \dots, m$. It is a $N \times N$ matrix with the input vector \mathbf{W} , and the output vector corresponding to the observations at the receiver r_i .

$$\mathbf{M}(r_i) = \mathbf{A}(\mathbf{I} - \mathbf{\Gamma})^{-1}\mathbf{B}(r_i), \quad i = 1, 2, \dots, m, \quad (55)$$

where matrix \mathbf{A} is a source matrix, $\mathbf{\Gamma}$ is adjacency matrix of the multicast network, and $\mathbf{B}(r_i)$ is an output matrix. For more details on how these matrices are constructed, we refer the interested reader to the reference [21]. Here, we just make a comment on the source matrix \mathbf{A} . In general, it is a $N \times \ell$ matrix, where ℓ is the total number of edges in the network. Input to the matrix \mathbf{A} is the vector of independent packets \mathbf{W} . For the source model in Figure 8, non-zero entries in the matrix \mathbf{A} correspond to the edges $S - s_1$, $S - s_2$ and $S - s_3$. Since, transmissions on those edges are already assigned by the

underlying source model, in general we have

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^T & \mathbf{A}_2^T & \cdots & \mathbf{A}_m^T & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \quad (56)$$

where \mathbf{A}_i corresponds to the observation matrix defined in (51).

Essentially, a multicast problem has a network coding solution if and only if each matrix $\mathbf{M}(r_i)$ is non-singular. In [18], the author showed that for the *expanded transfer matrix* defined as

$$\mathbf{E}(r_i) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} - \mathbf{\Gamma} & \mathbf{B}(r_i) \end{bmatrix}, \quad i = 1, 2, \dots, m, \quad (57)$$

it holds that $\det(\mathbf{M}(r_i)) = \pm \det(\mathbf{E}(r_i))$.

It should be noted that some of the entries in matrices $\mathbf{\Gamma}$ and $\mathbf{B}(r_i)$, $i = 1, 2, \dots, m$, are unknowns. To obtain the actual transmissions on all the edges it is necessary to replace those unknown entries with elements over \mathbb{F}_q such that all matrices $\mathbf{E}(r_i)$, $i = 1, 2, \dots, m$, have full rank. This is known as a simultaneous matrix completion problem and it is solved in [18] in polynomial time.

Lemma 9 (Harvey, [18]). *Polynomial time solution for the simultaneous matrix completion problem exists if and only if $|\mathbb{F}_q| > m$. The complexity of the proposed algorithm applied to the data exchange problem is $\mathcal{O}(m^4 \cdot N^3 \cdot n^3 \cdot \log(m \cdot N \cdot n))$.*

The complexity of the code construction can be further reduced when for the $(R_1, R_2, \dots, R_m) \in \mathcal{R}_n(\mathbf{1})$ it holds that the greatest common divisor $\gcd(nR_1, nR_2, \dots, nR_m) > 1$. In this case, for every $\tilde{n} = \frac{n}{\gcd(nR_1, nR_2, \dots, nR_m)}$ generations of the finite linear process, we still have that each user transmits some integer number of symbols in \mathbb{F}_q . Hence, it is enough to construct a coding scheme for \tilde{n} observations of the linear process, and then just to apply such scheme $\frac{n}{\tilde{n}}$ times to solve the data exchange problem. From Lemma 9 the complexity of such scheme is $\mathcal{O}(m^4 \cdot N^3 \cdot \tilde{n}^3 \cdot \log(m \cdot N \cdot \tilde{n}))$.

C. Asymptotic optimality of $R_{CO,n}(\mathbf{1})$

In this section we consider under which conditions there is no gap between the solution of the problem $\text{ILP}_n(\mathbf{1})$, when n is finite, and the solution of $\text{LP}_1(\mathbf{1})$ (asymptotic solution $n \rightarrow \infty$). To that end, let us consider the following Lemma.

Lemma 10. *Optimal $R_{CO}(\mathbf{1})$ rate of the problem $\text{LP}_1(\mathbf{1})$ can be expressed as*

$$R_{CO}(\mathbf{1}) = H(X_{\mathcal{M}}) - \min_{\mathcal{P}} \left\{ \frac{\sum_{S \in \mathcal{P}} H(X_S) - H(X_{\mathcal{M}})}{|\mathcal{P}| - 1} \right\}, \quad \mathcal{P} \text{ is a partition of } \mathcal{M} \text{ s.t. } |\mathcal{P}| \geq 2. \quad (58)$$

Proof of Lemma 10 is provided in Appendix F. It is based on a geometry of the function $g(\mathcal{M}, \beta)$. Minimization (58) was also shown in [11] by considering an LP dual of the optimization problem $\text{LP}_1(\mathbf{1})$.

From Lemma 10, $R_{CO}(\mathbf{1})$ can be expressed as a rational number. Moreover, the denominator of $R_{CO}(\mathbf{1})$ can be some integer number between 1 and $m - 1$ depending on the cardinality of the optimal partition according to (58). From Lemma 3 it immediately follows that all $(R_1, R_2, \dots, R_m) \in \mathcal{R}_n(\mathbf{1})$ are also rational numbers with the denominator n .

To that end, if n is divisible by $|\mathcal{P}(R_{CO}(\mathbf{1})) - 1|$ for $|\mathcal{P}(R_{CO}(\mathbf{1}))| \geq 2$, then

$$R_{CO,n}(\mathbf{1}) = R_{CO}(\mathbf{1}). \quad (59)$$

D. Achieving a rate tuple from $\mathcal{R}_n(\underline{\alpha})$

In Section VI-B we argued that once we obtain the optimal fractional rates (which denote how many symbols in \mathbb{F}_q each user transmits), the construction of the corresponding multicast network is straightforward, and hence, the coding scheme can be obtained in polynomial time by using the algorithm proposed in [18]. Here, we describe an algorithm that finds an optimal solution to the optimization problem $\text{ILP}_n(\underline{\alpha})$.

In Section V we proposed the gradient descent algorithm to achieve an approximate solution to the problem $\text{LP}_1(\underline{\alpha})$. Setting the precision parameter $\varepsilon = \frac{1}{n}$ it is guaranteed that the distance between the sum rate which corresponds to the rate tuple from $\mathcal{R}(\underline{\alpha})$ and the sum rate obtained through the gradient descent algorithm, is at most $\frac{1}{n}$, i.e., $|\beta_{gd} - \beta^*| \leq \frac{1}{n}$. Therefore, we have

$$|n\beta_{gd} - n\beta^*| \leq 1. \quad (60)$$

From (60) we conclude that

$$\left| \frac{\lfloor n\beta_{gd} \rfloor}{n} - \beta^* \right| \leq \frac{1}{n} \quad \text{or} \quad \left| \frac{\lceil n\beta_{gd} \rceil}{n} - \beta^* \right| \leq \frac{1}{n}. \quad (61)$$

From (61) it follows that we can achieve a rate tuple from $\mathcal{R}_n(\underline{\alpha})$ which sum rate is at most $\frac{1}{n}$ away from β^* by choosing $\beta = \frac{\lfloor n\beta_{gd} \rfloor}{n}$ or $\beta = \frac{\lceil n\beta_{gd} \rceil}{n}$. Let us denote by $\beta_{(n)}$ the optimal sum rate w.r.t. $\text{ILP}_n(\underline{\alpha})$. To decide which one of the proposed β 's is equal to $\beta_{(n)}$, we just need to compare the values of the function h at these points.

$$\beta_{(n)} = \arg \min \left\{ h\left(\frac{\lfloor n\beta_{gd} \rfloor}{n}\right), h\left(\frac{\lceil n\beta_{gd} \rceil}{n}\right) \right\}. \quad (62)$$

Then, it follows that

$$|R_{CO,n}(\underline{\alpha}) - R_{CO}(\underline{\alpha})| \leq \frac{\max_i \alpha_i}{n}. \quad (63)$$

Complexity of the proposed algorithm is $\mathcal{O}(m^2 \cdot SFM(m) + \log(n \cdot N) \cdot m \cdot SFM(m))$. After obtaining an optimal communication rates w.r.t. $ILP_n(\underline{\alpha})$, it is only left to apply the code construction algorithm proposed in Subsection VI-A (see Lemma 9).

E. Asymptotic optimality of $R_{CO,n}(\underline{\alpha})$

In this section we explore under which conditions the optimal solutions of the problems $ILP_n(\underline{\alpha})$ and $LP_1(\underline{\alpha})$ are the same.

In order to obtain the asymptotically optimal rates w.r.t. $LP_1(\underline{\alpha})$, it is necessary to bound from below the length of each linear segment in $h(\beta)$. Then, by choosing the appropriate step size in the gradient descent algorithm, we can achieve the goal.

Theorem 7. *An optimal asymptotic solution to the problem $LP_1(\underline{\alpha})$ in the finite linear source model can be obtained in polynomial time by using a gradient descent method with the precision parameter $\varepsilon = m^{-m/2}$. Complexity of the proposed algorithm is $\mathcal{O}((m \cdot \log m + \log N) \cdot m \cdot SFM(m))$.*

Proof: In Theorem 5 we showed that each breakpoint in $h(\beta)$ corresponds to a vertex of the rate region \mathcal{R} defined in (3). In other words, for some breakpoint β_j , the optimal rate tuple is uniquely defined by the following system of equations

$$R(\mathcal{S}_i) = H(X_{\mathcal{S}_i} | X_{\mathcal{S}_i^c}), \quad i = 1, 2, \dots, m, \quad (64)$$

where $\mathcal{S}_i \subset \mathcal{M}$. Moreover, it holds that $R(\mathcal{M}) = \beta_r$. System of linear equations (64) can be expressed in a matrix form as follows.

$$\mathbf{\Lambda} \cdot \mathbf{R} = \begin{bmatrix} H(X_{\mathcal{S}_1} | X_{\mathcal{S}_1^c}) & H(X_{\mathcal{S}_2} | X_{\mathcal{S}_2^c}) & \dots & H(X_{\mathcal{S}_m} | X_{\mathcal{S}_m^c}) \end{bmatrix}^T, \quad (65)$$

where

$$\Lambda_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{S}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (66)$$

In order to obtain the optimal rate tuple which corresponds to the breakpoint β_r , we can simply invert the matrix $\mathbf{\Lambda}$. Notice that the right hand side of (65) consists of the conditional entropy (rank) expressions, which are, in the case of the linear source model, integers. Therefore, all optimal rates \mathbf{R} which correspond to the breakpoints of $h(\beta)$ are fractional numbers with the denominator equal to the $\det(\mathbf{\Lambda})$. This comes from the fact that

$$\mathbf{\Lambda}^{-1} = \frac{1}{\det(\mathbf{\Lambda})} \cdot \text{adj}(\mathbf{\Lambda}), \quad (67)$$

where $\text{adj}(\mathbf{\Lambda})$ is the adjugate of $\mathbf{\Lambda}$. From [23] it follows that

$$|\det(\mathbf{\Lambda})| \leq m^{m/2}. \quad (68)$$

Therefore, all the breakpoints of $h(\beta)$ are at the distance of at least $m^{-m/2}$ from each other. Hence, by setting the precision parameter in a gradient descent algorithm to $\varepsilon = m^{-m/2}$, we can make sure that the minimizer of $h(\beta)$ is the end point of the linear segment to which approximate solution belongs to. ■

In the further text, we explain how to find the minimum of $h(\beta)$ by applying a simple binary search algorithm on top of the gradient descent algorithm proposed in Theorem 7. Let us consider the scenario in Figure 9. Applying the gradient descent algorithm, with the precision parameter $\varepsilon = m^{-m/2}$ we can reach a point β_{gd} that is ε close to the minimizer β^* of $h(\beta)$, i.e., $|\beta_{gd} - \beta^*| \leq m^{-m/2}$. Applying the modified Edmond's algorithm for $\beta = \beta_{gd}$, we obtain parameters $(\mathbf{b}^{(gd)}, \mathbf{c}^{(gd)})$ (see (44)) which correspond to the linear segment to which β_{gd} belongs to. In order to obtain β^* we simply need to jump to the consecutive linear segment. To that end, let $\beta_1 = \beta_{gd} - m^{-m/2}$ belongs to the linear segment $(\mathbf{b}^{(1)}, \mathbf{c}^{(1)})$. Then, β^* can be obtained by intersecting these two linear segments.

$$\beta^* = \frac{\sum_{i=1}^m c_i^{(1)} \alpha_i - \sum_{i=1}^m c_i^{(gd)} \alpha_i}{\sum_{i=1}^m b_i^{(gd)} \alpha_i - \sum_{i=1}^m b_i^{(1)} \alpha_i}. \quad (69)$$

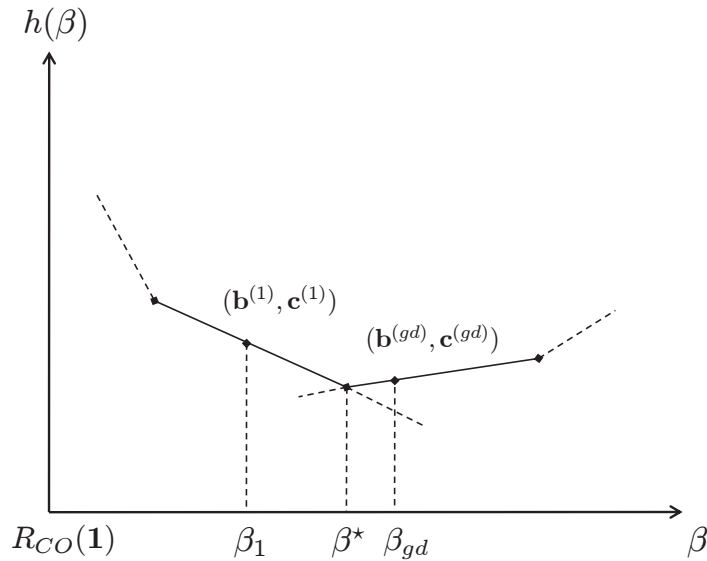


Fig. 9. Line intersection procedure applied on top of the gradient descent algorithm to obtain the minimum $h(\beta^*)$.

Therefore, if the data packet length n is divisible by the denominator of β^* , then $R_{CO,n}(\underline{\alpha}) = R_{CO}(\underline{\alpha})$.

F. Indivisible Packets

Let us now consider the scenario when the data packets cannot be split. To obtain an optimal communication rates, we can directly apply the results from the Sections VI-A and VI-D. We can think of this problem as having one packet over very large base field \mathbb{F}_{q^n} .

Hence, for the case when $\underline{\alpha} = \mathbf{1}$, it holds that

$$R_{CO,1}(\mathbf{1}) = \lceil R_{CO}(\mathbf{1}) \rceil \text{ symbols in } \mathbb{F}_{q^n}.$$

Similarly, we can obtain the sum rate which corresponds to the optimal $R_{CO,1}(\underline{\alpha})$ as follows.

$$\beta_{(1)} = \arg \min \{h(\lfloor \beta_{gd} \rfloor), h(\lceil \beta_{gd} \rceil)\} \text{ symbols in } \mathbb{F}_{q^n}.$$

However, in the actual coding scheme, all the algebraic operations are performed over the original base field \mathbb{F}_q .

VII. CONCLUSION

In this work we addressed the problem of the data exchange, where each user in the system possess some partial knowledge (side information) about the file that is of common interest. The goal is for each user to gain access to the entire file while minimizing the (possibly weighted) amount of bits that these users need to exchange over a noiseless public channel. For the general case when the side information is in form of the i.i.d. realizations of some discrete memoryless process, we provide a polynomial time algorithm that finds an optimal rate allocation w.r.t. communication cost. Our solution is based on some combinatorial optimization techniques such as optimizations over submodular polyhedrons, Dilworth truncation of intersecting submodular functions, Edmond's greedy algorithm, etc. For the case when the side information is in form of the linearly coded packets, besides an optimal rate allocation in polynomial time, we provide efficient methods for constructing linear network codes that can achieve omniscience among the users at the optimal rates with finite block lengths and zero-error.

APPENDIX A

PROOF OF LEMMA 1

Base polyhedron $B(f, \leq)$ is defined by the following system of inequalities

$$Z(\mathcal{S}) \leq f(\mathcal{S}), \quad \mathcal{S} \subset \mathcal{M}, \tag{70}$$

$$Z(\mathcal{M}) = f(\mathcal{M}). \tag{71}$$

This is equivalent to the following

$$Z(\mathcal{S}^c) \geq f^*(\mathcal{S}^c) (= f(\mathcal{M}) - f(\mathcal{S})), \quad (72)$$

$$Z(\mathcal{M}) = f^*(\mathcal{M}) (= f(\mathcal{M})), \quad (73)$$

where the last equality holds because $f(\emptyset) = 0$. For the second part, we have

$$\begin{aligned} (f^*)^*(\mathcal{S}) &= f^*(\mathcal{M}) - f^*(\mathcal{S}^c) \\ &= f(\mathcal{M}) - (f(\mathcal{M}) - f(\mathcal{S})) = f(\mathcal{S}). \end{aligned}$$

APPENDIX B

PROOF OF LEMMA 4

Using the properties of conditional entropy, we can write $f(\mathcal{S}, \beta) = \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}})$. When $\mathcal{S} \cap \mathcal{T} \neq \emptyset$, then the following inequality holds due to submodularity of entropy

$$\begin{aligned} f(\mathcal{S}, \beta) + f(\mathcal{T}, \beta) &= H(X_{\mathcal{S}}) + H(X_{\mathcal{T}}) - 2(H(X_{\mathcal{M}}) - \beta) \\ &\geq H(X_{\mathcal{S} \cup \mathcal{T}}) + H(X_{\mathcal{S} \cap \mathcal{T}}) - 2(H(X_{\mathcal{M}}) - \beta) = f(\mathcal{S} \cup \mathcal{T}, \beta) + f(\mathcal{S} \cap \mathcal{T}, \beta). \end{aligned} \quad (74)$$

Inequality (74) holds whenever $\mathcal{S} \cap \mathcal{T} \neq \emptyset$. To show that the function f is submodular when $\beta \geq H(X_{\mathcal{M}})$ it is only left to consider the case $\mathcal{S} \cap \mathcal{T} = \emptyset$. Since $f(\emptyset, \beta) = 0$, we have

$$\begin{aligned} f(\mathcal{S}, \beta) + f(\mathcal{T}, \beta) &= H(X_{\mathcal{S}}) + H(X_{\mathcal{T}}) - 2(H(X_{\mathcal{M}}) - \beta) \\ &\geq H(X_{\mathcal{S}}, X_{\mathcal{T}}) - (H(X_{\mathcal{M}}) - \beta) = f(\mathcal{S} \cup \mathcal{T}, \beta). \end{aligned} \quad (75)$$

Inequality in (75) follows from the fact that

$$H(X_{\mathcal{S}}) + H(X_{\mathcal{T}}) - H(X_{\mathcal{S}, \mathcal{T}}) = I(X_{\mathcal{S}}; X_{\mathcal{T}}) \geq 0 \geq \beta - H(X_{\mathcal{M}}). \quad (76)$$

This completes the proof.

APPENDIX C

PROOF OF LEMMA 5

Let us define function $g(\mathcal{M}, \beta, i)$, $i = 1, 2, \dots, m$ as follows

$$g(\mathcal{M}, \beta, i) = \min_{\mathcal{P}} \left\{ \sum_{\mathcal{S} \in \mathcal{P}} \beta - H(X_{\mathcal{S}^c} | X_{\mathcal{S}}), \quad \text{s.t. } |\mathcal{P}| = i : \mathcal{P} \text{ is a partition of } \mathcal{M} \right\}. \quad (77)$$

Function $g(\mathcal{M}, \beta, i)$ is linear in β for any fixed $i = 1, 2, \dots, m$. Then, the Dilworth truncation $g(\mathcal{M}, \beta)$ can be written as

$$g(\mathcal{M}, \beta) = \min_{i=1,2,\dots,m} g(\mathcal{M}, \beta, i). \quad (78)$$

Note that the minimization (77) does not depend on β since it can be written as

$$g(\mathcal{M}, \beta, i) = i(\beta - H(X_{\mathcal{M}})) + \min_{\mathcal{P}} \left\{ \sum_{S \in \mathcal{P}} H(X_S), \text{ s.t. } |\mathcal{P}| = i : \mathcal{P} \text{ is a partition of } \mathcal{M} \right\}. \quad (79)$$

Therefore, $g(\mathcal{M}, \beta)$ can be solved for any given β by minimizing over all m lines $g(\mathcal{M}, \beta, i)$, $i =$

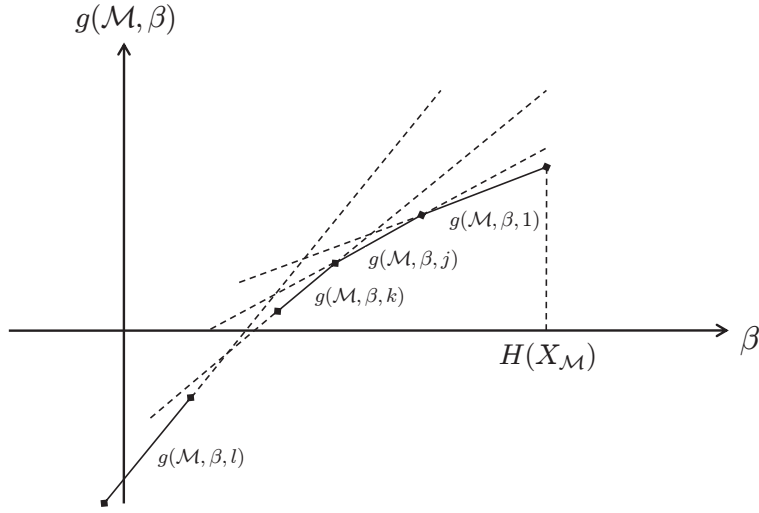


Fig. 10. Function $g(\mathcal{M}, \beta)$ is piecewise linear in β . It can be obtained by minimization over m linear functions $g(\mathcal{M}, \beta, i)$, $i = 1, 2, \dots, m$. $g(\mathcal{M}, \beta)$ has non-increasing slope, i.e., $1 \leq j \leq k \leq \dots \leq l \leq m$.

$1, 2, \dots, m$. Hence, $g(\mathcal{M}, \beta)$ has at most m linear segments. Moreover, due to minimization (78), $g(\mathcal{M}, \beta)$ has non-increasing slope (see Figure 10).

To verify that the last linear segment in $g(\mathcal{M}, \beta)$ is of slope 1, it is sufficient to find a point β for which the function $g(\mathcal{M}, \beta)$ has slope 1. To that end, let us consider $\beta = H(X_{\mathcal{M}})$. From Lemma 4 it follows that $f(\mathcal{S}, \beta = H(X_{\mathcal{M}}))$ is submodular function, and hence, $g(\mathcal{M}, \beta = H(X_{\mathcal{M}})) = f(\mathcal{M}, \beta = H(X_{\mathcal{M}})) = \beta$, where the last equality follows from (29). Therefore, the slope of $g(\mathcal{M}, \beta)$ at $\beta = H(X_{\mathcal{M}})$ is 1, which completes the proof.

APPENDIX D

OPTIMAL PARTITIONING W.R.T. DILWORTH TRUNCATION

In [16] it was shown how to obtain an optimal partition $\mathcal{P}(\beta)$ of the set \mathcal{M} w.r.t. (34) from the modified Edmond's algorithm. Here we provide intuition behind these results. From Remark 3 it follows that $g(\mathcal{M}, \beta)$ is the optimal value of the optimization problem $\text{LP}_2(\beta)$. As we pointed out in Section III, in each iteration i of the modified Edmond's algorithm, we obtain a set \mathcal{S}_i for which the inequality constraint in $P(f, \beta)$ holds with equality. In the next claim we state a result that is crucial for obtaining an optimal partition of \mathcal{M} with respect to Dilworth truncation of $f(\mathcal{M}, \beta)$.

Claim 2. *For an optimal solution \mathbf{Z} of the problem $\text{LP}_2(\beta)$, if $Z(\mathcal{S}_1) = f(\mathcal{S}_1)$, and $Z(\mathcal{S}_1) = f(\mathcal{S}_2)$ then $Z(\mathcal{S}_1 \cup \mathcal{S}_2) = f(\mathcal{S}_1 \cup \mathcal{S}_2)$.*

Proof: For an optimal rate vector \mathbf{Z} of the problem $\text{LP}_2(\beta)$ we have

$$Z(\mathcal{S}_i) = \beta - H(X_{\mathcal{S}_i^c} | X_{\mathcal{S}_i}) = \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}), \quad (80)$$

$$Z(\mathcal{S}_j) = \beta - H(X_{\mathcal{S}_j^c} | X_{\mathcal{S}_j}) = \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_j}). \quad (81)$$

Since $\text{LP}_2(\beta)$ represents optimization over the polyhedron $P(f, \beta)$ it holds that

$$Z(\mathcal{S}_i \cup \mathcal{S}_j) \leq \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}, X_{\mathcal{S}_j}), \quad (82)$$

$$Z(\mathcal{S}_i \cap \mathcal{S}_j) \leq \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i \cap \mathcal{S}_j}). \quad (83)$$

From (80) and (81) it follows that

$$\begin{aligned} Z(\mathcal{S}_i \cup \mathcal{S}_j) &= Z(\mathcal{S}_i) + Z(\mathcal{S}_j) - Z(\mathcal{S}_i \cap \mathcal{S}_j) \\ &= \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}) + \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_j}) - Z(\mathcal{S}_i \cap \mathcal{S}_j) \\ &\geq \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}) + H(X_{\mathcal{S}_j}) - H(X_{\mathcal{S}_i \cap \mathcal{S}_j}), \end{aligned} \quad (84)$$

where the last step in (84) follows from (83). Due to submodularity of entropy it directly follows from (84) that

$$Z(\mathcal{S}_i \cup \mathcal{S}_j) \geq \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}, X_{\mathcal{S}_j}). \quad (85)$$

Comparing (82) and (85) it must hold that

$$Z(\mathcal{S}_i \cup \mathcal{S}_j) = \beta - H(X_{\mathcal{M}}) + H(X_{\mathcal{S}_i}, X_{\mathcal{S}_j}). \quad (86)$$

■

Results of Claim 2 represent a key building block for obtaining an optimal partition $\mathcal{P}(\beta)$ for some fixed β (see Algorithm 3). From Remark 3 it follows that for the maximizer rate vector \mathbf{Z} of the problem $\text{LP}_2(\beta)$ it holds that

$$Z(\mathcal{S}) = f(\mathcal{S}, \beta), \quad \forall \mathcal{S} \in \mathcal{P}(\beta). \quad (87)$$

From Claim 2 and (87) it follows that for the sets \mathcal{S}_i and \mathcal{S}_j , which are the minimizer sets in iterations i and j of the modified Edmond's algorithm, if $\mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$, then $\mathcal{S}_i \cup \mathcal{S}_j$ is a subset of the one of the partition sets in $\mathcal{P}(\beta)$. Therefore, in each iteration of the modified Edmond's algorithm, whenever the minimizer set intersects some of the previously obtained sets, they must all belong to the same partition set (see steps 4 and 5 in Algorithm 3). Algorithm 3 compared to the modified Edmond's Algorithm,

Algorithm 3 Optimal Partition [16]

- 1: Let $j(1), j(2), \dots, j(m)$ be any ordering of $\{1, 2, \dots, m\}$, and $\mathcal{A}_i = \{j(1), j(2), \dots, j(i)\}$.
- 2: Initialize $\mathcal{P}^0 = \emptyset$.
- 3: **for** $i = 1$ to m **do**
- 4: Let \mathcal{S}_i be the minimizer of

$$Z_{j(i)} = \min\{f(\mathcal{S}, \beta) - Z(\mathcal{S}) : j(i) \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{A}_i\}.$$

- 5: $\mathcal{T}_i = \mathcal{S}_i \cup [\cup\{\mathcal{V} : \mathcal{V} \in \mathcal{P}^{i-1}, \mathcal{S}_i \cap \mathcal{V} \neq \emptyset\}]$
 - 6: $\mathcal{P}^i = \{\mathcal{T}_i\} \cup \{\mathcal{V} : \mathcal{V} \in \mathcal{P}^{i-1}, \mathcal{S}_i \cap \mathcal{V} = \emptyset\}$
 - 7: **end for**
 - 8: $\mathcal{P}(\beta) = \mathcal{P}^m$.
-

has two additional steps in each iteration (step 5 and step 6). Thus, the order of complexity of both algorithms is the same and it is $\mathcal{O}(m \cdot \text{SF}M(m))$. The complete explanation of the Algorithm 3 can be found in [16].

APPENDIX E

PROOF OF LEMMA 6

Function $h(\beta)$ is given by

$$h(\beta) = \min_{\mathbf{R}} \sum_{i=1}^m \alpha_i R_i \quad \text{s.t.} \quad R(\mathcal{M}) = \beta, \quad R(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subset \mathcal{M}. \quad (88)$$

Continuity of $h(\beta)$

Let the rate tuple $(R_1^{(1)}, R_2^{(1)}, \dots, R_m^{(1)})$ corresponds to the minimizer β_1 of the function $h(\beta)$, i.e., $\sum_{i=1}^m R_i^{(1)} = \beta_1$. Then, for a point $\beta_2 = \beta_1 + \Delta\beta$ let us construct the rate tuple

$$R_i^{(2)} = \begin{cases} R_i^{(1)} + \Delta\beta & \text{if } \beta = 1, \\ R_i^{(1)} & \text{if } \beta \neq 1. \end{cases} \quad (89)$$

Then $(R_1^{(2)}, R_2^{(2)}, \dots, R_m^{(2)})$ is a feasible rate tuple for the optimization problem (88) when $\sum_{i=1}^m R_i^{(2)} = \beta_2$. Moreover, it holds that $h(\beta_2) - h(\beta_1) \leq \alpha_1 \Delta\beta$. Hence,

$$|\beta_2 - \beta_1| \leq \Delta\beta \Rightarrow |h(\beta_2) - h(\beta_1)| \leq \alpha_1 \Delta\beta, \quad (90)$$

Since $\alpha_1 < \infty$ by the model assumption, it immediately follows that the function $h(\beta)$ is continuous.

Convexity of $h(\beta)$

Consider two points β_1 and β_2 such that $\beta_i \geq R_{CO}(\mathbf{1})$, $i = 1, 2$. We want to show that for any $\lambda \in [0, 1]$ it holds that $h(\lambda\beta_1 + (1 - \lambda)\beta_2) \leq \lambda h(\beta_1) + (1 - \lambda)h(\beta_2)$. To that end, let $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ be the optimal rate tuples w.r.t. $h(\beta_1)$ and $h(\beta_2)$, respectively. Now, we show that $\mathbf{R} = \lambda\mathbf{R}^{(1)} + (1 - \lambda)\mathbf{R}^{(2)}$ is feasible rate tuple for the problem (88) when $\beta = \lambda\beta_1 + (1 - \lambda)\beta_2$.

Since $R^{(1)}(\mathcal{M}) = \beta_1$ and $R^{(2)}(\mathcal{M}) = \beta_2$, it follows that

$$R(\mathcal{M}) = \lambda R^{(1)}(\mathcal{M}) + (1 - \lambda)R^{(2)}(\mathcal{M}) = \lambda\beta_1 + (1 - \lambda)\beta_2. \quad (91)$$

Since $R^{(1)}(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c})$, $R^{(2)}(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c})$, $\forall \mathcal{S} \subset \mathcal{M}$, we have

$$R(\mathcal{S}) = \lambda R^{(1)}(\mathcal{S}) + (1 - \lambda)R^{(2)}(\mathcal{S}) \geq H(X_{\mathcal{S}}|X_{\mathcal{S}^c}). \quad (92)$$

From (91) and (92) it follows that \mathbf{R} is a feasible rate tuple w.r.t. optimization problem (88). Therefore, $\sum_{i=1}^m \alpha_i R_i \geq h(\lambda\beta_1 + (1 - \lambda)\beta_2)$. Hence,

$$h(\lambda\beta_1 + (1 - \lambda)\beta_2) \leq \lambda h(\beta_1) + (1 - \lambda)h(\beta_2), \quad (93)$$

which completes the proof.

APPENDIX F
PROOF OF LEMMA 10

For $\beta = R_{CO}(\mathbf{1})$ it holds that $|\mathcal{P}(\beta)| = 1$. Since $\beta = R_{CO}(\mathbf{1})$ is also a breakpoint in $g(\mathcal{M}, \beta)$ (see Lemma 5), we have that $|\mathcal{P}(\beta)| \geq 2$. In other words, optimal partition of the set \mathcal{M} is not unique. From (34) and (35) we can write expression for $R_{CO}(\mathbf{1})$ as follows

$$R_{CO}(\mathbf{1}) = |\mathcal{P}(R_{CO}(\mathbf{1}))| R_{CO}(\mathbf{1}) - \sum_{S \in \mathcal{P}(R_{CO}(\mathbf{1}))} H(X_{S^c} | X_S). \quad (94)$$

Rearranging terms in (94) we get

$$(|\mathcal{P}(R_{CO}(\mathbf{1}))| - 1) R_{CO}(\mathbf{1}) = \sum_{S \in \mathcal{P}(R_{CO}(\mathbf{1}))} H(X_S) - |\mathcal{P}(R_{CO}(\mathbf{1}))| H(X_{\mathcal{M}}). \quad (95)$$

Dividing both sides of equality by $(|\mathcal{P}(R_{CO}(\mathbf{1}))| - 1)$ we obtain

$$R_{CO}(\mathbf{1}) = H(X_{\mathcal{M}}) - \frac{\sum_{S \in \mathcal{P}(R_{CO}(\mathbf{1}))} H(X_S) - H(X_{\mathcal{M}})}{|\mathcal{P}(R_{CO}(\mathbf{1}))| - 1}. \quad (96)$$

This completes the proof of (58) since $|\mathcal{P}(R_{CO}(\mathbf{1}))| \geq 2$.

APPENDIX G
PROOF OF THEOREM 6

We prove this theorem by showing that for any rate tuple \mathbf{R} that belongs to the rate region \mathcal{R}_{de} , defined in (9), there exists a network coding solution to the data exchange problem.

In the data exchange problem, each of the m users get to observe some collection of linear combinations of the data packets w_1, w_2, \dots, w_N .

$$X_i = \mathbf{A}_i \cdot \mathbf{W}, \quad \forall i \in \mathcal{M}, \quad (97)$$

where $\mathbf{A}_i \in \mathbb{F}_p^{\ell_i \times N}$, and $\mathbf{W} = \begin{bmatrix} w_1 & w_2 & \dots & w_N \end{bmatrix}^T \in \mathbb{F}_p^N$.

Since each user is interested in recovering all the data packets \mathbf{W} , one can convert the data exchange problem into a multicast network problem. For instance, considering the user 1 as a receiver (see Figure 11), it obtains the side information from himself (thus the link of capacity ℓ_1 from user 1 to user 1), and it receives transmissions from the other users through the links of capacities R_i , $i = 2, 3, \dots, m$. But, in order to set up the problem this way it is necessary to know how many symbols in \mathbb{F}_{q^n} each user broadcasts, *i.e.*, we need to know the capacities R_i of the links.

In [22], the authors proved necessary and sufficient conditions for the existence of the network coding solution when the sources are linearly correlated. In the following Lemma we state their result adapted

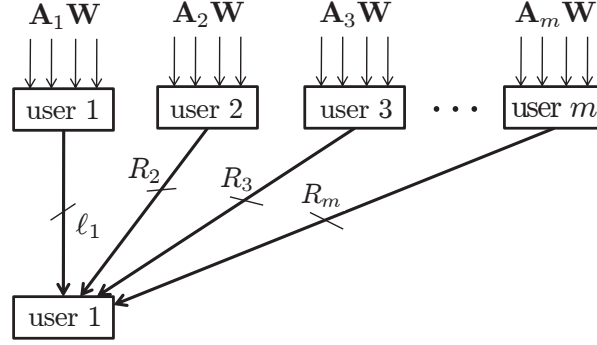


Fig. 11. Data exchange problem can be interpreted as a multicast problem. Considering user 1 as a receiver, it obtains the side information from himself through the link of capacity ℓ_1 , and it receives transmissions from the other users through the links of capacities R_i , $i = 2, 3, \dots, m$.

to the data exchange problem with linearly coded packets. Let us denote by $\mathbf{A}_j(\mathcal{S}_i, \star)$ a sub-matrix of \mathbf{A}_j with rows indexed by the elements of the set \mathcal{S}_i .

Lemma 11. *In the data exchange problem with linearly coded packets, a rate tuple (R_1, R_2, \dots, R_m) can be achieved by network coding if and only if*

$$\text{rank}(\mathbf{A}_1, \mathbf{A}_2(\mathcal{S}_2^{(1)}, \star), \dots, \mathbf{A}_m(\mathcal{S}_m^{(1)}, \star)) = N, \quad (98)$$

$$\text{rank}(\mathbf{A}_1(\mathcal{S}_1^{(2)}, \star), \mathbf{A}_2, \dots, \mathbf{A}_m(\mathcal{S}_m^{(2)}, \star)) = N, \quad (99)$$

$$\vdots$$

$$\vdots$$

$$\text{rank}(\mathbf{A}_1(\mathcal{S}_1^{(m)}, \star), \dots, \mathbf{A}_{m-1}(\mathcal{S}_{m-1}^{(m)}, \star), \mathbf{A}_m) = N, \quad (100)$$

such that $|\mathcal{S}_i^{(j)}| = R_i$, $\forall j \in \{1, 2, \dots, m\}$, $\forall i \in \{1, 2, \dots, m\} \setminus \{j\}$, where $\mathcal{S}_i^{(j)} \subseteq \{1, 2, \dots, \ell_i\}$.

Each equation in (98)-(100) corresponds to the selection of N disjoint paths from the users 1 through m , to one of the receiving users (see Figure 11 where user 1 is the receiving user). Hence, for a rate tuple (R_1, R_2, \dots, R_m) that satisfies the conditions in Lemma 11 there exists a network coding solution to the data exchange problem. Now, let us consider the equations (98) through (100). The idea is to identify the set of all achievable solutions for each receiver, *i.e.*, the goal is to find the collection of sets $\{\mathcal{S}_i^{(j)}\}_{i=1, i \neq j}^m$ for each $j \in \{1, 2, \dots, m\}$ which satisfy the conditions of the j^{th} row in (98)-(100). To that end let us consider Algorithm 4 (see [12]).

Algorithm 4 Greedy Algorithm

```

1: Initialize  $j(1) \in \{1, 2, \dots, m\}$ ,  $\mathbf{S} = \mathbf{A}_{j(1)}$ .
2: Let  $j(2), j(3), \dots, j(m)$  be any ordering of  $\{1, 2, \dots, m\} \setminus \{j(1)\}$ .
3: for  $i = 2$  to  $m$  do
4:   Initialize  $\mathcal{S}_{j(i)}^{(j(1))} = \emptyset$ .
5:   for  $k = 1$  to  $\ell_{j(i)}$  do
6:     if  $\text{rank}(\mathbf{S}, \mathbf{A}_{j(i)}(k, \star)) = \text{rank}\{\mathbf{S}\} + \text{rank}\{\mathbf{A}_{j(i)}(k, \star)\}$  then
7:        $\mathbf{S} = \begin{bmatrix} \mathbf{S} \\ \mathbf{A}_{j(i)}(k, \star) \end{bmatrix}$ ,  $\mathcal{S}_{j(i)}^{(j(1))} = \mathcal{S}_{j(i)}^{(j(1))} \cup \{k\}$ .
8:     end if
9:   end for
10: end for

```

It is not hard to conclude that Algorithm 4 satisfies the maximum rank property, *i.e.*, for every $j(1) \in \{1, 2, \dots, m\}$ it holds that

$$\begin{aligned} & \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}(\mathcal{S}_{j(2)}^{(j(1))}, \star), \dots, \mathbf{A}_{j(i)}(\mathcal{S}_{j(i)}^{(j(1))}, \star)) \\ &= \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(i)}), \quad i = 2, 3, \dots, m \end{aligned} \quad (101)$$

Therefore, for one particular ordering $j(1), j(2), \dots, j(m)$ of $1, 2, \dots, m$, we have that

$$R_{j(i)} = \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(i)}) - \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(i-1)}), \quad i = 2, 3, \dots, m. \quad (102)$$

From (102) it follows that

$$\begin{aligned} \sum_{i=t}^m R_{j(i)} &= \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(m)}) - \text{rank}(\mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(t-1)}) \\ &= \text{rank}(\mathbf{A}_{j(t)}, \mathbf{A}_{j(t+1)}, \dots, \mathbf{A}_{j(m)} | \mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(t-1)}), \quad t = 2, 3, \dots, m. \end{aligned} \quad (103)$$

Since the feasibility condition has to be satisfied for any ordering, we conclude that if for every ordering $j(1), j(2), \dots, j(m)$ of $1, 2, \dots, m$

$$\sum_{i=t}^m R_{j(i)} \geq \text{rank}(\mathbf{A}_{j(t)}, \mathbf{A}_{j(t+1)}, \dots, \mathbf{A}_{j(m)} | \mathbf{A}_{j(1)}, \mathbf{A}_{j(2)}, \dots, \mathbf{A}_{j(t-1)}), \quad t = 2, 3, \dots, m, \quad (104)$$

then (R_1, R_2, \dots, R_m) can be achieved by network coding. It is not hard to see that the rate region in (104) is equivalent to

$$\sum_{i \in \mathcal{S}} R_i \geq \text{rank}(\mathbf{A}_{\mathcal{S}} | \mathbf{A}_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subset \{1, 2, \dots, m\}. \quad (105)$$

Thus, we showed that the cut-set bounds (105) for the data exchange problem with linearly coded packets can be achieved via network coding.

REFERENCES

- [1] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *Proceedings of ITW*, 2010.
- [2] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *Proceedings of ISIT*, 2010, pp. 1888–1892.
- [3] T. Courtade, B. Xie, and R. Wesel, "Optimal Exchange of Packets for Universal Recovery in Broadcast Networks," in *Proceedings of Military Communications Conference*, 2010.
- [4] S. Tajbakhsh, P. Sadeghi, and R. Shams, "A model for packet splitting and fairness analysis in network coded cooperative data exchange."
- [5] D. Ozgul and A. Sprintson, "An algorithm for cooperative data exchange with cost criterion," in *Information Theory and Applications Workshop (ITA)*, 2011. IEEE, pp. 1–4.
- [6] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3047–3061, 2004.
- [7] J. Orlin, "A faster strongly polynomial time algorithm for submodular function minimization," *Mathematical Programming*, vol. 118, no. 2, pp. 237–251, 2009.
- [8] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," *Information Theory, IEEE Transactions on*, vol. 49, no. 3, pp. 626–643, 2003.
- [9] —, "Generalized coset codes for distributed binning," *Information Theory, IEEE Transactions on*, vol. 51, no. 10, pp. 3457–3474, 2005.
- [10] V. Stankovic, A. Liveris, Z. Xiong, and C. Georgiades, "On code design for the slepian-wolf problem and lossless multiterminal networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1495–1507, 2006.
- [11] C. Chan, "Generating Secret in a Network," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.
- [12] A. Schrijver, *Combinatorial optimization*. Springer, 2003.
- [13] S. Fujishige, *Submodular functions and optimization*. Elsevier Science, 2005.
- [14] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," *Combinatorial structures and their applications*, pp. 69–87, 1970.
- [15] S. Fujishige and N. Tomizawa, "A note on submodular functions on distributive lattices," *Journal of the Operations Research Society of Japan*, vol. 26, pp. 309–318, 1983.
- [16] K. Nagano, Y. Kawahara, and S. Iwata, "Minimum Average Cost Clustering," *Advances in Neural Information Processing Systems*, vol. 23, 2010.
- [17] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.
- [18] N. Harvey, D. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 489–498.
- [19] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [20] D. Bertsekas, *Network optimization: Continuous and discrete methods*. Athena Scientific (Belmont, Mass.), 1998.
- [21] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782 – 795, 2003.

- [22] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [23] J. Hadamard, “Résolution d’une question relative aux déterminant,” *Bull. Sci. Math*, vol. 17, pp. 240–246, 1893.